BOEK & CASSETTE

BASICODE-3



Kluwer Technische Boeken

Het BASICODE-3 boek

Het BASICODE-3 boek

Dit boek kwam tot stand onder auspiciën van de
Stichting BASICODE.

Bijdragen voor de tekst werden geleverd door: Hermine Bakker, Jan Bredenbeek, Yoeri Geutskens, Jacques Haubrich, Jochem Herrmann, Henk Koorn, Peter Maathuis, Harm Mulder, Klaas Robers, Henk Warnitz, Peter Zevenhoven.

De eindredactie was in handen van Jacques Haubrich.

De cassette werd geproduceerd in een studio van het Philips Natuurkundig Laboratorium met technische medewerking van de heer W. van Duynhoven.

De verbindende teksten op de cassette werden ingesproken door Henny Obbes.



KLUWER TECHNISCHE BOEKEN B.V.
DEVENTER-ANTWERPEN

ISBN 90 201 1949 4
D/1986/0108/206

1e druk 1986

Woord vooraf

Dit BASICODE-3 pakket is een zeer opmerkelijke uitgave. Ten eerste omdat dit het gezamenlijke werk is van een flink aantal mensen. Ten tweede omdat BASICODE een soort 'Esperanto' voor thuiscomputers is. Daarmee is het namelijk mogelijk in BASIC computerprogramma's te maken die zonder meer op diverse merken computers zijn te gebruiken.

BASICODE bestaat al weer heel wat jaartjes. Dit nieuwe BASICODE-3 is een sterk uitgebreide en verbeterde versie. Dat heeft de BASICODE-hobbyisten-programmeurs vele avonden vrije tijd gekost. Het is goed dat u weet dat daar weinig anders tegenover staat dan het genoegen van een gezamenlijk stuk werk. Hun namen vindt u verderop in dit boek. Zij maakten BASICODE-3 mogelijk, een standaard die ongetwijfeld, net als zijn voorganger BASICODE-2, binnen en buiten Nederland zijn gebruikers zal weten te vinden.

Omdat het belangrijk is de kwaliteit van BASICODE te bewaken, is de STICHTING BASICODE in het leven geroepen. Deze Stichting bundelt de samenwerking tussen de auteurs van vertaalprogramma's en is voortgekomen uit de BASICODE-groep van het eerste uur. Het hoofddoel van de Stichting is de bevordering van het gebruik van BASICODE. Daarbij hoort de bewaking van de kwaliteit. Niemand is gebaat bij slecht werkende programma's. Dat geldt natuurlijk voor de BASICODE-vertaalprogramma's maar in zekere mate ook voor de programma's geschreven in BASICODE. Wie een vertaalprogramma aan het maken is voor zijn eigen merk thuiscomputer moet vast en zeker contact opnemen met de Stichting BASICODE, voor informatie, hulp en afstemming. In voorkomende gevallen kan de hulp van de Stichting ook worden gevraagd voor screening tegen de standaard van programma's in BASICODE.
Het adres:
Stichting BASICODE,
Postbus 1410,
5602 BK Eindhoven.

Ik wens u veel genoegen bij het gebruik van deze unieke Nederlandse ontwikkeling.

Klaas Robers, voorzitter Stichting BASICODE, april 1986

Inhoud

1 BASICODE, wat is dat?

Wie net zijn of haar eerste schreden in computerland zet, heeft het niet gemakkelijk. Er worden daar nogal wat bijzondere woorden gebruikt en er zijn zoveel dingen waar je op moet letten. En dan al die programma's: toen je het voor het eerst zag, leek het allemaal zo gemakkelijk. Bandje erin, draaien en hup, daar verschijnen de eerste beelden al op het scherm. Maar heb je eenmaal zelf een computer, dan blijkt het ineens veel minder vanzelfsprekend te zijn. Voor het ene programma moet het geheugen worden uitgebreid, voor het andere is een andere interpreter nodig. Waarom maakt men dat toch allemaal zo ingewikkeld? En dan al die verschillende computertalen. Net ben je een beetje gewend aan BASIC, want dat was het helemaal volgens het boekje dat bij de computer zat, en je hoort overal zeggen dat PASCAL veel beter is, terwijl anderen ineens blijken te zweren bij programma's in machinetaal.

Natuurlijk is het zo dat alle begin moeilijk is. Maar er is meer. Door de snelle ontwikkeling van de techniek en de mogelijkheid deze nieuwe technieken meteen toe te passen in de thuiscomputers van nu, is er een doorlopende verandering te zien. De computer van vandaag is gebouwd met de nog warme techniek van gisteren. Morgen is er een redelijk aantal programma's voor te koop, maar er is dan ook al weer een nieuwe computer. Een nieuwere computer kan meestal iets meer. En dan is er een probleem, want in de programmeertaal van de bestaande modellen is daar niet op gerekend. Snel wordt er een aantal nieuwe BASIC-opdrachten verzonnen en toegevoegd aan de bestaande lijst van instructies. En dat gebeurt natuurlijk niet op één plaats in de wereld. Zo kan het gebeuren dat er in de lijst van BASIC-instructies van verschillende merken thuiscomputers nogal wat verschillen voorkomen.

Dat is helaas nog niet alles. Wie ooit heeft geprobeerd een cassette met een programma van een ander soort thuiscomputer in te lezen, weet dat dat niet gaat. Dat komt niet alleen door de verschillen in de BASIC-instructies. Draai de bandjes maar eens af op een gewone cassetterecorder. Wat je dan hoort, klinkt anders. Het gefluit en geratel heeft een andere toon. Daar weet je eigen computer geen raad mee. Het uitwisselen van programma's tussen verschillende computers is daardoor vrijwel onmogelijk geworden. Zou dat misschien met opzet zo gemaakt zijn?

BASICODE lost een aantal van deze problemen voor u op. Daartoe zijn drie stappen genomen.

1. In veel programma's is het niet nodig gebruik te maken van de bijzondere mogelijkheden van een bepaald soort thuiscomputer. BASICODE geeft een soort

'Grootste Gemene Deler' aan van BASIC-instructies. Een lijst van opdrachten die elke BASIC-interpreter wel aankan. Een programma dat alleen hiervan gebruik maakt, kan dus in vrijwel elke computer werken.

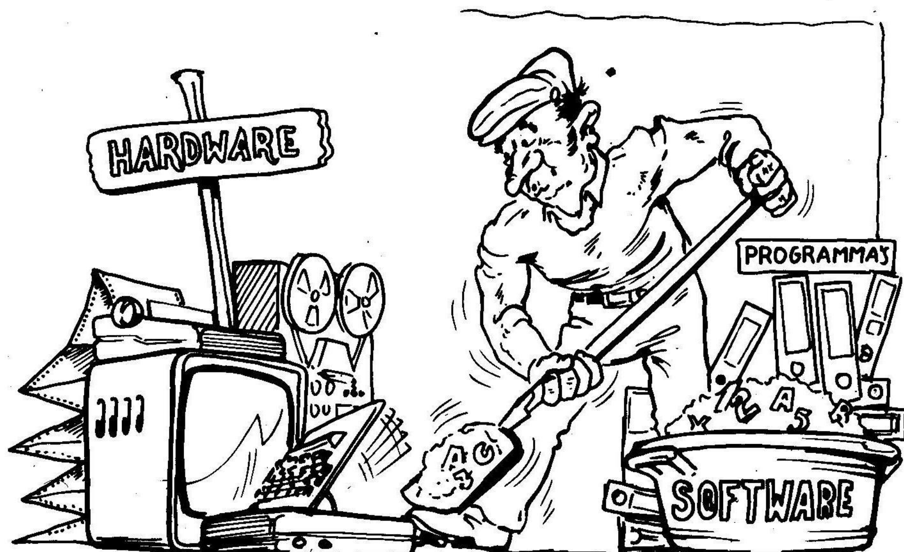
2. Daarna blijkt er nog een aantal handelingen te zijn die je in een programma haast niet kunt missen. Een voorbeeld dat iedereen zal aanspreken, is zoiets eenvoudigs als het schoonmaken van het beeldscherm. Dat gaat in bijna iedere computer anders. BASICODE geeft een eenvoudige omweg aan, voor kenners: een afgesproken subroutine, waarmee de handeling voor alle computers op dezelfde manier in het programma kan worden verwerkt.

3. Het wegschrijven op een cassette en het teruglezen van de programma's gebeurt in vrijwel elke computer door een stukje programma in machinetaal. De aansluiting van de cassetterecorder op de microprocessor, het hart van de computer, is zo eenvoudig en goedkoop mogelijk gehouden. Alle slimmigheidjes zitten in het machinetaalprogramma. Dat zit altijd al in de computer, voor de kenners: in het ROM. Hierin worden ook de toontjes en ratelgeluiden bepaald waarmee de BASIC-programma's op de cassette worden gezet. Een ander programma voor de besturing van de cassetteaansluiting en er wordt een ander geluid opgenomen. Voor BASICODE is er eerst precies afgesproken wat voor toontjes er gebruikt moeten worden. Daarna is er voor elke computer een nieuw stukje machinetaal gemaakt, zodat er gelezen en geschreven kan worden met deze afgesproken tonen. Dit programma maakt deel uit van het BASICODE-vertaalprogramma. Dat is verschillend voor elk soort thujscomputer. Het moet worden ingelezen net zoals een gewoon programma van die computer. Maar daarna is het mogelijk om een programma in BASICODE van een cassette in te lezen. Dat kan dus uit een heel andere computer komen, want wat er op de cassette staat, is van te voren heel precies afgesproken. Zo is het meestal ook meteen mogelijk een eigen BASIC-programma op de cassette te zetten op deze BASICODE-manier.

Een set BASIC-instructies, die op elke computer werken. Een stel afgesproken subroutines voor speciale handelingen. Een vaste manier om op cassettes te schrijven. Dat alles in een hulpprogramma per computermerk. Dat is BASICODE. In het onderwijs, met op elke school een ander merk computer, is het niet meer weg te denken.

In dit boek staat alles over BASICODE. Over hoe het ontstaan is en naderhand in stappen uitgebreid. De officiële richtlijnen voor het maken van programma's, die zonder problemen op elke computer lopen. In vakkringen heet dat 'het protocol'. Op de bij dit boek behorende cassette staan de vertaalprogramma's voor een groot aantal soorten in Nederland gangbare thuiscomputers. Hoe die precies gebruikt moeten worden, staat verderop in dit boek. Daarbij komen zoveel mogelijk de mensen aan het woord die de vertaalprogramma's zelf hebben gemaakt. Wie kan het beter vertellen dan zij? En dan enkele programma's, geschreven volgens het BASICODE-protocol, uitgebreid toegelicht. Zo krijgt u snel een indruk van hoe het in de praktijk werkt en waarop gelet moet worden. Achterin het boek, als naslagwerk, staan alle belangrijke afspraken en regels van BASICODE. Heel in het kort vindt u datzelfde nog eens op de los bijgevoegde overzichtskaart.

Wij hopen dat u dit boek met aandacht bekijkt. Het resultaat van dit gezamenlijke project van een aantal enthousiaste computerhobbyisten is het zeker waard. En verder hopen wij dat u veel plezier beleeft aan de universele programma's in BASICODE. Misschien dat er daar binnenkort ook bijkomen die door u gemaakt zijn. De voldoening is groot om na enige tijd uw eigen programma terug te zien, perfect lopend op een op dit moment misschien nog onbekende computer.



2 De ontwikkeling van BASICODE

BASICODE is eigenlijk ontstaan door een radioprogramma. In 1979 is de NOS in het radioprogramma Hobbyscoop gestart met het uitzenden van computerprogramma's. Het was begonnen als een proefje, maar al gauw ging Hans Janssen, presentator van dit programma voor hobbyisten, ertoe over wekelijks een computerprogramma uit te zenden. Afwisselend was dit voor een andere van een viertal op dat moment redelijk verspreide computers, de Apple II, de Exidy, de PET (Commodore) en de TRS 80 model 1 (Tandy). Er werd elke week een door hobbyisten ingestuurd programma gesave-d op studiotape. Dit werd aan het einde van het radioprogramma ingelast in de uitzending. De luisteraars werd aangeraden alles op de cassetterecorder op te nemen en naderhand het computerprogramma op de gebruikelijke manier in de eigen computer in te lezen. De uitzendingen werden verzorgd via het kwalitatief uitstekende FM-zendernet. Bij een storingsvrije ontvangst ging dat heel redelijk. En zo hoorden de niets vermoedende luisteraars elke week een paar minuten lang onbegrijpelijke en afschuwelijke geluiden uit hun radio komen.

Er waren eigenlijk twee grote problemen. Ten eerste moesten de luisteraars zorgvuldig het roulatieschema van de computermerken in de gaten houden. Eens per vier weken kwam een bepaalde computer aan de beurt. Als die uitzending vergeten werd, moest er weer vier weken gewacht worden voor het volgende inleesbare computerprogramma kwam. Daarbij bleven de hobbyisten met andere merken computers in de kou staan. Een verdere versnippering was ook in niemands belang. Ten tweede bleek dat als er ook maar éven wat storing was tijdens de uitzending van het computerprogramma het hele programma geweigerd werd. Dat komt door de manier waarop een programma in de vorm van toontjes en pulsjes op de cassette wordt gezet. Bij een verkeerde keuze hiervan zal een korte onderbreking of een extra puls de rest van het programma ernstig verminken. Toch wordt deze rest wel goed overgezonden. Bij alle merken computers ontbrak echter de mogelijkheid om met goed verstand naderhand nog iets te verbeteren. Ging er iets fout, dan was er helemaal geen programma. Een zuiver binair (tweewaardig) gegeven, dat wel, maar voor enthousiaste hobbyisten niet te verteren.

Hans Janssen was door zijn uitzendingen in contact gekomen met een aantal zeer actieve computerhobbyisten. Van hen kreeg hij ook vele van de programma's die werden uitgezonden. De problemen die aan deze manier van uitzenden kleven, waren natuurlijk bij allen bekend. Het was begin 1981 dat één van hen, Klaas Robers, ingenieur op het Natuurkundig Laboratorium van Philips, met een nieuw idee kwam. Daarmee moest het mogelijk zijn de computerprogramma's zó uit te zenden dat alle merken

ze konden inlezen. Het was de grondgedachte van BASICODE: een nauwkeurig afgesproken toontjescode om programma's op een cassette te zetten en een bijbehorend hulpprogramma per computersoort. Door een geschikt gekozen codering zou een programma dat met wat storing wordt ontvangen, slechts minimaal verminkt zijn. Elke computerhobbyist in die dagen had voldoende kennis van programmeren om deze verminkingen te kunnen verhelpen.

Op de avond van donderdag 14 mei 1981 werd een aantal actieve hobbyisten bij elkaar getrommeld om het idee te bespreken. Elk van hen kende zijn eigen computer van binnen en van buiten. Dat was nodig, want niemand had in zijn eentje de kennis om voor een andere dan zijn eigen machine de behoorlijk ingewikkelde machinetaalroutines te maken. BASICODE kon alleen wat worden door een gezamenlijke aanpak. Op deze avond, in een lokaaltje van het NOS-opleidingscentrum Santbergen, ontvouwde Klaas Robers zijn plannen. Kort tevoren was er een uitzending geweest met daarin een kort programma in een voorlopige versie van wat Klaas dacht dat het zou kunnen worden. De resultaten hiervan waren zó bemoedigend dat gezamenlijk besloten werd er mee door te gaan. Tijdens de bespreking van de voorstellen werden enige zeer zinvolle toevoegingen gedaan in de 'standaard'. Na afloop ging ieder naar huis, met in het hoofd een gemeenschappelijk doel: BASICODE!

De hieropvolgende weken waren vol van spanning en druk programmeren. Er werd een cassettebandje rondgestuurd met een eenvoudig testprogramma. BASICODE was in het begin nog niet meer dan een afgesproken manier om BASIC-programma's op een cassette te zetten. De dingen die anders gingen in de verschillende computers zouden naderhand gemakkelijk in het ontvangen programma aangepast kunnen worden. Het feit alleen al dat men elkaars programma's zou kunnen inlezen was een opwindende gedachte. En daaraan werd gewerkt in de vrije tijd en de zomervakantie van 1981.

Op 11 augustus had de NOS een persconferentie gepland om BASICODE wereldkundig te maken. Dit was de eerste keer dat men weer bij elkaar kwam en ieder had zijn eigen computer meegenomen. En zie, het wonder geschiedde. Een cassette, geschreven op het ene merk, werd aan een ander gegeven en kon zomaar worden ingelezen. Bij iedereen hoorde je dezelfde ratelende geluiden, ongeacht de vorm, kleur en naam van de computer. Dit was waar voor gewerkt was. Maar als je het zag en hoorde gebeuren, was het fantastischer dan je je had voorgesteld. Door een gemeenschappelijke inspanning was de taalbarrière overwonnen.

Eind september 1981 werd BASICODE in de pers geïntroduceerd en werden de eerste vertaalprogramma's via de radio uitgezonden. Na korte tijd waren er meer dan tien merken waarvoor in enige vorm een BASICODE-vertaalprogramma gemaakt was. Er werden soms zeer creatief gemaakte programma's in BASICODE uitgezonden. Niet onvermeld mag blijven het kerstprogramma van Hermine Bakker. Dit speelde de muziek "Stille Nacht, Heilige Nacht" op een groot aantal verschillende merken. Dat het daarvoor nodig was voor elk merk een afzonderlijke muziekroutine te maken, en dat deze allemaal in het programma aanwezig waren, dat zagen alleen de kenners. De moeite die Hermine heeft genomen om deze routines bij de andere leden van de groep los te krijgen, was kenmerkend voor het enthousiasme.

Maar er moest nog veel werk verzet worden. Het thuiscomputeren won aan populariteit en er moest hoognodig iets komen voor de luisteraar die pas een computer had gekocht na de uitzending van het vertaalprogramma van zijn merk. In een nieuwe onderlinge samenwerking van de BASICODE-groep, de NOS en de Hobby Computer Club (HCC) werd een boekwerk samengesteld met daarbij een cassette waarop alle vertaalprogramma's stonden. De teksten waren behalve in het Nederlands ook in het Engels afgedrukt, vertaald door Jonathan Marks, medewerker van de Wereldomroep. Bijna een jaar na de persconferentie bracht de NOS dit boek uit als een BASICODE-beginset voor haar luisteraars. In een paar maanden was de eerste oplage van 2000 stuks uitverkocht.

Het werd ondertussen tijd eens te kijken naar wat er verder te doen stond. In de praktijk werkte de gebruikte toontjescode probleemloos. De uitzendingen waren overal goed te ontvangen en door het zelfherstellend vermogen was een kleine storing geen probleem. Het moeilijkste was nog wel het aanpassen van de uitgezonden programma's aan het eigen merk. Door de popularisatie van de thuiscomputer was de gemiddelde gebruiker minder goed bekend met het programmeren. De handreiking naar de gebruiker zou eigenlijk verder moeten gaan. Was er een mogelijkheid het zó te maken dat de programma's zonder meer konden lopen op alle aangesloten computers?

Op de avond van 30 september 1982, weer in gebouw Santbergen te Hilversum, kwam de ondertussen uitgebreide en wat gewijzigde BASICODE-groep bijeen om nieuwe plannen te maken. Klaas Robers had samen met Jochem Herrmann, toen nog student aan de TH Eindhoven, plannen uitgewerkt om te komen tot een programmastructuur met vaste subroutines. In het eerste jaar BASICODE was er een goed inzicht ontstaan in de grootste knelpunten. Bepaalde handelingen moesten uitgevoerd kunnen worden in de programma's. Bovendien bleken er enkele instructies in BASIC door de verschillende computers wat anders te worden behandeld, zoals het op het scherm zetten van getallen. De voorgestelde methode met vast gedefinieerde subroutines kon de bestaande problemen oplossen. Gezamenlijk werden de regelnummers gekozen voor de standaard-subroutines. Ook leek het verstandig wat meer vastigheid te geven in de regelnummering binnen het programma. Speciaal uit de hoek van het onderwijs was hierom gevraagd, zodat men wat gemakkelijker de weg zou kunnen vinden in een nieuw ontvangen computerprogramma. Om het onderscheid duidelijk te laten blijken, werd gekozen om voortaan te spreken over BASICODE-2 wanneer de subroutines nodig zijn voor het lopen van het programma. De oude versie, zonder deze uitbreiding, zou blijven bestaan onder de naam BASICODE-1. Hiermee bleef het mogelijk programma's ongewijzigd over te spelen van het ene merk naar het andere.

Het werd ook hoog tijd om eens een overzicht te maken van het doen en laten van de verschillende computertypen. Hierdoor ontstond er een lijst van de BASIC-instructies die overal hetzelfde doen. Het overgrote deel van de computers werkte met een BASIC van Microsoft, daardoor was er vrij veel overeenkomst. Er werd door Jochem een protocol samengesteld. Hierin stonden de afgesproken subroutines en de gemeenschappelijke BASIC-instructies nauwkeurig omschreven. Het protocol werd, samen met een cassette met een testprogramma, aan alle leden van de groep toegestuurd. Het programma, een zichzelf aan de schermafmetingen aanpassende uitleg van BASICODE-2, gebruikte alle beschreven subroutines.

Ondertussen werd er ook gewerkt aan een nieuw BASICODE-boek. In plaats van een herdruk van het uitverkochte BASICODE-boek kwam er in juli 1983 meteen een BASICODE-2 boek uit, natuurlijk weer met een cassette. Het protocol stond volledig in dit boek. Ongeveer tegelijk met het verschijnen van het nieuwe boek werd BASICODE-2 via Hobbyscoop geïntroduceerd. De belangstelling was zeer groot. Voor het eerst was het echt mogelijk programma's te maken die zonder meer op verschillende merken computers liepen. Het onderwijs was laaiend enthousiast. De beperkingen die BASICODE-2 oplegde, wogen veel minder zwaar dan het voordeel van de uitwisselbaarheid van de programma's.

Ook in het buitenland begon de belangstelling te groeien. Aangemoedigd door de NOS begonnen de WDR en de BBC met uitzendingen van BASICODE. In Duitsland ging dat op het geluidskanaal van de TV. De BBC begon op de lange golf met uitzendingen midden in de nacht op radio-4; later verhuisde BASICODE naar het Engelse Radio-1 FM-zendernet. In de praktijk bleek de overdracht via middengolf en lange golf in de avond en nacht onbruikbaar voor de snelle BASICODE-signalen. Ook de korte golf is hiervoor niet te gebruiken. Jonathan Marks deed proeven via de Wereldomroep die dit bevestigden. Radio-zendamateurs merkten dat de VHF- en UHF-banden geknipt waren voor onderlinge uitwisseling van digitale informatie via BASICODE.

Ook de pers ontwikkelde belangstelling voor BASICODE-2. In computertijdschriften verschenen listings van programma's in BASICODE-2 en er kwamen zelfs complete boekjes van. In Duitsland verscheen een beginnerspakket bij de grote uitgever Ravensburger. In deze uitgave waren echter enkele programma's opgenomen waarin stukjes voorkwamen met computerspecifieke opdrachten. Eén van deze programma's was een tekstverwerker. Daarbij was het natuurlijk nodig de teksten te bewaren en terug te roepen. Bij een ander programma werd kleur gebruikt. Beide zaken waren niet voorzien in BASICODE-2.

Op 11 mei 1985 kwam de BASICODE-groep weer bijeen in een zaaltje in één van de Philips-complexen te Eindhoven. Het was iedereen duidelijk: BASICODE-3 moest er komen. Een aantal lang begeerde zaken moest mogelijk worden gemaakt in deze uitbreiding. Jochem Herrmann en Jacques Haubrich, leraar wiskunde te Eindhoven, hadden een uitgebreid voorstel gemaakt. Muziek, grafische mogelijkheden, inverse video, bestanden, er was heel wat bij gekomen. Het zou niet gemakkelijk worden, maar we waren ook veel wijzer geworden in de loop der tijd.

En weer ging de groep uit elkaar met een gemeenschappelijk doel: BASICODE nog beter maken. In vele honderden uren programmeren ontstonden nieuwe, uitgebreide vertaalprogramma's. U hebt het resultaat nu in uw handen: BASICODE-3. Wat het gaat worden, zal de toekomst leren. Het ontstaan en de ontwikkeling zijn het werk van een aantal zeer enthousiaste hobbyisten die erg veel van hun vrije tijd besteed hebben aan een doel dat alleen gezamenlijk bereikt kon worden.

3 Hoe gebruik je BASICODE-3

3.1 Inleiding

BASICODE maakt het dus mogelijk programma's zó op cassette te schrijven dat praktisch elke microcomputer in staat is de cassette te lezen en er een normaal werkend programma van te maken. Het BASICODE-vertaalprogramma zorgt daarvoor. Het vervangt in feite het door de fabrikant in de computer ingebouwde cassette-leessysteem. In dit hoofdstuk behandelen we de hoofdzaken daarvan, met name bezien vanuit de vraag hoe u zelf daarmee overweg kunt.

3.2 Wat u moet doen

Een cassette met een BASICODE-opname is in geen enkele computer zonder meer in te lezen. De reden is heel simpel: elke computerfabrikant houdt er zijn eigen methode op na hoe een programma op cassette wordt geschreven. Bij iedere fabrikant is dat anders. Luister de bij dit boek behorende cassette maar eens af, dan hoort u de enorme verschillen. Doordat het in uw computer ingebouwde cassette-leesprogramma alleen cassettes kan lezen die zijn beschreven met geluiden volgens het systeem dat de fabrikant van uw computer heeft bedacht, kan dat programma meteen ook niets anders lezen. BASICODE is weer iets anders, dus dat kan niet worden gelezen. De oplossing is: schakel het door de fabrikant ingebouwde leesprogramma uit en gebruik een ander leesprogramma.

Dat andere leesprogramma staat op de cassette die bij dit boek hoort. De kunst is natuurlijk het leesprogramma voor uitgerekend uw eigen computer te vinden. Leg dus de cassette in een gewone audiorecorder en laat hem afspelen. U hoort telkens een kort stukje gesproken tekst. Daarin wordt vermeld voor welke computer het volgende programma is. Wacht rustig af tot uw computermerk wordt genoemd. Stop dan de audiorecorder en leg de cassette in de recorder van uw computer. Nu moet u vanaf die cassette het speciale programma inlezen. Met welke opdracht dat precies gaat, kunt u vinden in hoofdstuk 5, waar de details met betrekking tot uw computer worden vermeld. In de meeste gevallen gaat het met vrijwel dezelfde opdracht waarmee u een willekeurig ander programma van cassette inleest.

Na enige tijd is het programma ingelezen. In de meeste computers moet u dan een startopdracht intikken, bijvoorbeeld RUN. De juiste opdracht vindt u in hoofdstuk 5. Dan is het enkele ogenblikken stil. Het zojuist ingelezen programma schakelt het in uw computer ingebouwde leesprogramma uit en schakelt zichzelf daarvoor in de plaats. Als ook dat gebeurd is, is uw computer klaar om met BASICODE te werken.

3.3 Een BASICODE-programma inlezen

U kunt nu een BASICODE-programma inlezen. Dat gaat meestal NIET met de opdracht waarmee u normaal een programma inleest, immers die opdracht schakelt het normale leessysteem in. U moet nu juist het BASICODE-leessysteem inschakelen. Dat gaat met een speciale, afwijkende opdracht. Welke opdracht? Dat vindt u weer in de beschrijving in hoofdstuk 5. Leg tevoren een cassette met een BASICODE-opname in de recorder. U kunt om te beginnen de B-kant van de bij dit boek behorende cassette daarvoor gebruiken; daarop staan namelijk enkele programma's in BASICODE. Via de luidspreker van een normale cassetterecorder kunt u de aankondiging daarvan horen. Leg de cassette zó in de computerrecorder dat bij het afspelen de start van de pieptoon het eerste is wat uw computer te horen krijgt. Geef dan de speciale leesopdracht die u in hoofdstuk 5 kunt vinden. Het BASICODE-leesprogramma zal nu de reutelgeluiden van de band inlezen en ze omzetten in een programma dat in uw computer kan werken. Tijdens het inlezen zult u op het scherm op een of andere manier kunnen volgen dat er iets wordt ingelezen. Als het einde van de opname is bereikt, kan het nog even duren voordat u zelf op een toets mag drukken. De meeste BASICODE-leesprogramma's hebben dan nog wat tijd nodig om alles wat is ingelezen om te zetten in een normaal, goed werkend programma. Ook dat omzetproces kunt u op een of andere manier op het scherm volgen.

3.4 Een BASICODE-programma gebruiken

Na afloop van het inleesproces is het grote moment gekomen. Is het programma goed ingelezen? Is de omzetting naar een normaal programma goed gelopen? Gelukkig is meestal op beide vragen het antwoord JA! In dat geval zijn er geen problemen.

Het kan echter gebeuren dat de in te lezen band niet door een andere computer is beschreven, maar via grote omwegen, zoals een radio- of TV-zender en een andere cassetterecorder bij u terecht is gekomen. Al die omwegen kunnen storingen in de opname veroorzaken, waardoor het inlezen misschien niet helemaal foutloos verloopt. BASICODE is zodanig gemaakt dat ondanks eventuele fouten toch al datgene wat wél goed is overgekomen ook inderdaad goed in de computer is opgenomen. Als er dus wordt gemeld dat er leesfouten zijn opgetreden, dan kunt u met LIST-opdrachten bekijken waar er misschien iets mis is en - aangenomen dat u een beetje kijk op programmeren hebt - in veel gevallen de opgetreden fouten herkennen en meestal zelfs verbeteren.

Is, zoals het hoort, alles foutloos ingelezen, dan komt nu het grote moment: werkt het 'vreemde' programma inderdaad in uw computer? Geef daartoe een RUN-opdracht (of de daarvoor benodigde andere instructie, zie hoofdstuk 5) en kijk wat er gebeurt. Als alles gaat zoals het hoort, en dat is gelukkig meestal zo, dan doet het programma van alles. Of u het inderdaad leuk en nuttig vindt, is een andere zaak. Niet iedereen hoeft hevig geïnteresseerd te zijn in de sterrenhemel. Maar toch, het wonder is geschied: een op een ander merk computer ontwikkeld programma is in uw computer ingelezen en blijkt nog te werken ook!

3.5 Wat doe je met dat programma?

Tenzij het ingelezen programma u absoluut niet interesseert, zult u het voor later gebruik willen opslaan. Dat kan op diskette (als u een schijfeenheid hebt) of op cassette. Hoe het programma daarop wordt weggeschreven, is meestal weer in elke computer anders, maar in veel gevallen is het een SAVE-opdracht. Eenmaal weggeschreven met die opdracht is het programma alleen nog maar door uw eigen computer en soortgelijke exemplaren in te lezen. Tenzij....-, ja tenzij u het op de BASICODE-manier weer op cassette laat opnemen. Dat kan. Het BASICODE-vertaalprogramma kan namelijk niet alleen een programma inlezen, het kan ook een programma volgens het BASICODE-systeem op cassette schrijven. Dat is handig als dat programma in een computer van een ander merk moet worden ingelezen.

3.6 Wegschrijven in BASICODE

Voordat we een programma wegschrijven even dit. Computerprogramma's zijn allemaal door iemand gemaakt. Die auteur is de eigenaar van dat programma en die auteur is de enige die zijn (of haar) eigen produkt mag kopiëren. Zonder diens toestemming mag u het programma alleen zelf gebruiken, maar mag u er geen kopie van aan iemand anders geven, laat staan aan iemand anders verkopen. Dat geldt voor elk BASICODE-vertaalprogramma zoals dat op bijgaande cassette staat en dat geldt even zeer voor elk BASICODE-programma. Het nu volgende dient u in dit licht te lezen. Voor eigen gebruik mag u elk programma dat u op legale wijze in bezit hebt gekregen zo vaak kopiëren als u wilt. Maar alléén voor eigen gebruik.

Om te beginnen zou u er goed aan doen het vertaalprogramma voor uw computertype te kopiëren op een eigen cassette (of diskette). U hoeft dan niet meer elke keer op de BASICODE-cassette te zoeken waar het programma staat. Hoe u van het vertaalprogramma een kopie maakt, vindt u in hoofdstuk 5.

Hoe je een gewoon BASIC-programma volgens het BASICODE-systeem op cassette zet, staat eveneens in hoofdstuk 5. Het is van het grootste belang, daarvoor exact de juiste commando's in te tikken, immers, de geproduceerde opname moet ook door andere computers gelezen kunnen worden! Daarom zijn er voor dat wegschrijven een groot aantal regels en de meeste vertaalprogramma's controleren bij het wegschrijven op enkele of vele van die regels. De exacte regels staan in hoofdstuk 4 vermeld: het zogenaamde BASICODE-protocol. De belangrijkste daarvan, met name die regels waar de meeste fouten tegen worden gemaakt, worden nu even doorgenomen.

BASICODE eist dat een programmaregel nooit langer is dan 60 tekens. Sommige computers komen namelijk in moeilijkheden als ze langere regels binnen krijgen. Het kan dus zijn dat het vertaalprogramma in uw computer het wegschrijven in BASICODE weigert, omdat er in het programma één of meer regels voorkomen die te lang zijn.

Een ander aspect is het feit dat in BASICODE alleen de programmaregels met de regelnummers vanaf 1000 worden weggeschreven. De regels tot 1000 bevatten namelijk de BASICODE-subroutines. Die zijn helemaal aangepast aan uw eigen computer,

zijn dus voor een ander merk machine onbruikbaar en worden daarom bij het inlezen van BASICODE door het leesprogramma toegevoegd.

Bepaalde tekens en opdrachten mogen in BASICODE niet gebruikt worden. Opdrachten als POKE, CALL, SYS en USR (om maar een paar dwarsstraten te noemen) zijn absoluut verboden en hetzelde geldt voor de grafische tekentjes die bij diverse computers onder de toetsen zitten. Veel vertaalprogramma's weigeren dan het programma volgens de BASICODE-regels op cassette te schrijven. Gelukkig maar, want een andere computer zou er misschien de grootste moeilijkheden door krijgen.

In al deze gevallen moet u dus uw programma aanpassen. Het weg te schrijven programma dient eigenlijk voor 100% aan de regels van het BASICODE-protocol te voldoen. Dat is misschien wel eens moeilijk, maar met enige oefening en ervaring komt u een heel eind. Gelukkig is het zo dat veel BASICODE-programma's die officieel worden gepubliceerd, dus in tijdschriften, via centraal verspreide cassettes en via radio of TV tevoren op professionele wijze zijn gecontroleerd om te zien of ze inderdaad volledig aan het protocol voldoen. In die gevallen weet u zeker dat dat programma zonder problemen op uw computer zal werken. Als die controle niet gebeurd is, dan is de kans groot dat zo'n programma op minstens één merk of type computer problemen oplevert. Het is namelijk zó dat elk voorschrift dat in het protocol staat opgenomen daar staat, omdat in het verleden is gebleken dat er minstens één computer is die zonder die regel in moeilijkheden komt. Denk dus bij het lezen van het protocol niet 'wat een flauwekul' maar probeer juist de regels te onthouden en er rekening mee te houden. Alleen dán zal uw programma ook in alle andere computers werken.

3.7 Zelf programmeren in BASICODE

Na het voorafgaande zal het duidelijk zijn dat zelf programmeren in BASICODE wel enige discipline vraagt. Sommige opdrachten die u bij uw eigen programma's normaal kunt gebruiken, mogen in BASICODE niet worden toegepast. In plaats daarvan komt de aanroep van een standaard-subroutine. Het klassieke voorbeeld is het wissen van het scherm. Een Apple wist het beeld als u een HOME-opdracht geeft; bij een Commodore gaat het met PRINT en daarachter tussen aanhalingstekens een invers hartje en een BBC-computer reageert alleen op een CLS-opdracht. Zo is het bijna overal anders. In BASICODE wordt het scherm gewist met de opdracht GOSUB100. Probeert u het maar uit. Natuurlijk moet op regel 100 wel de juiste opdracht staan. Daar zorgt het vertaalprogramma voor. Zo zijn er in BASICODE een flink aantal vaste subroutines die allemaal een specifieke taak uitvoeren, die in de verschillende computers met telkens een andere opdracht wordt geprogrammeerd.

Daarnaast zijn er wat regels om bepaalde dingen te verbieden, maar dan wel zo dat u daar niet echt last van hebt. Een INPUT-opdracht mag in veel machines worden voorzien van een zogenaamde promptstring, een stukje tekst tussen aanhalingstekens dat vanzelf op het scherm komt te staan. Sommige computers kunnen daar niet goed mee overweg en daarom is vastgelegd dat je die tekst eerst zelf met PRINT moet laten afdrukken waarna een 'kale' INPUT-opdracht volgt. Zo zijn er meer beperkingen. Probeert u ze te onthouden en waar nodig toe te passen.

De derde en laatste groep regels betreft de regelnummers. Het is namelijk al heel wat, dat uw programma ook in andere computers kan worden ingelezen en meteen zal werken, maar het is nog fraaier als andere computeraars ook begrijpen hoe dat mooie programma in elkaar zit. Daarom zijn er enkele regels die vertellen welke regelnummers u voor welk doel mag gebruiken. Het spreekt vanzelf dat zo hier en daar een REM-opdracht geen kwaad kan, zeker niet als in die REM-opdracht een toelichting staat waarmee de bedoeling van het programma beter te begrijpen is.

Bestudeer daarom op uw gemak het volgende hoofdstuk. Doe dat over een of twee maanden nog eens en maak hier en daar een aantekening van die dingen waarvan u denkt dat u ze misschien erg gemakkelijk zou kunnen vergeten. Hoe beter u alle regels naleeft, des te meer plezier kunnen anderen van uw programma hebben.



4 Het BASICODE-3 protocol

4.1 Inleiding

Dit hoofdstuk bevat het officiële BASICODE-3 protocol. Dat protocol bevat alle regels waar men zich aan dient te houden. Een programma dat niet aan al deze regels voldoet, is dus geen goed BASICODE-3 programma. Wie een programma publiceert dat niet aan dit protocol voldoet, publiceert dus eigenlijk geen BASICODE-3 programma. Wie het dan toch BASICODE-3 noemt, jokt. Dat mag niet. Als iemand zoiets bij herhaling en met kennelijke opzet of uit pure nonchalance doet, dan riskeert hij het een en ander. BASICODE is een goed systeem en niemand is er bij gebaat dat anderen met slechte of onvolledige programma's de goede naam van BASICODE bederven. Eventuele klachten in dit opzicht vragen wij bij de uitgever van dit boek aan te melden.

4.2 Algemene afspraken

Zoals we reeds opmerkten, zijn er verschillen in de diverse uitvoeringen van de BASIC-programmeertaal. Om programma's niet alleen uit te kunnen wisselen maar ook zonder problemen op een ander type computer te kunnen laten werken, zijn in het BASICODE-3 protocol een aantal regels vastgelegd waaraan die programma's moeten voldoen:

a. We gebruiken alleen die BASIC-opdrachten die alle computers kennen en op dezelfde manier uitvoeren. Deze opdrachten worden gedetailleerd beschreven in paragraaf 4.6.

b. We reserveren de regelnummers tot regel 1000 voor subroutines die bepaalde belangrijke taken verrichten die op grond van regel a niet programmeerbaar zijn. De werking van deze subroutines is vastgelegd in paragraaf 4.3. Zo wist de met GOSUB 100 aangeroepen subroutine het beeldscherm, zet de subroutine op regel 110 de cursor op een bepaalde plaats op het scherm enz. Omdat deze routines voor elk type computer anders zijn, worden ze niet samen met het BASICODE-programma op cassette geschreven. Het is de taak van het vertaalprogramma deze routines toe te voegen en zo het programma compleet te maken.

c. De maten van het beeldscherm zijn bij vrijwel elk merk en type computer verschillend. Het aantal regels varieert van 16 (TRS-80) tot 32 of meer en het aantal tekens per regel is in een VIC-20 slechts 22 en in veel andere computers 40, 80 of soms nog meer. Zo lopen ook de grafische mogelijkheden sterk uiteen, van grove blokgrafiek met slechts 80 blokjes per regel tot redelijk hoge resolutie met 300 of

meer beeldpunten per lijn. Een goed BASICODE-programma houdt met zulke verschillen rekening. Bij de start van een BASICODE-3 programma staat daarom in de variabele HO hoeveel tekens per regel mogelijk zijn en in de variabele VE hoeveel regels er op het scherm kunnen (eigenlijk staat in beide variabelen 1 minder dan zojuist omschreven; we komen hier nog op terug). Evenzo wordt in de variabelen HG en VG opgegeven hoeveel grafische punten er horizontaal resp. verticaal in de desbetreffende computer mogelijk zijn.

Als redelijk gemiddelde mag men uitgaan van een beeldscherm van 24 regels met elk 40 tekens en een grafische resolutie van 180 lijnen met elk 250 beeldpunten. Een echt goed BASICODE-programma zorgt voor een schermindeling die overeenkomt met de mogelijkheden van de computer waar het in werkt.

d. Elke programmaregel mag inclusief het regelnummer en de spaties maximaal 60 tekens lang zijn.

e. Programma's die van een printer gebruik maken, dienen te rekenen op een printer met een regellengte van 80. Uitvoer naar de printer dient dus zodanig plaats te vinden dat nooit meer dan 80 tekens op dezelfde regel worden afgedrukt.

f. De programma-opbouw

Als het BASICODE-programma eenmaal compleet met subroutines in het geheugen staat, is de opbouw van het programma als volgt:

```
  1Ø GOTO 1ØØØ
  2Ø voorbereiding van de computer voor BASICODE
  25 GOTO 1Ø1Ø
 1ØØ machine-afhankelijke subroutines.
   I er is exact vastgelegd wat elke subroutine doet
 65Ø
   I
 95Ø terugzetten van de computer in normaal bedrijf
 955 END
1ØØØ bepaal stringruimte :GOTO2Ø
1Ø1Ø hier begint het BASICODE-programma
   I
   I BASICODE-programma
   I
.... laatste opdracht :GOTO 95Ø
```

Het programma begint en eindigt altijd in deze volgorde. Regel 10 is alleen bedoeld om over de standaard-subroutines heen te springen. Regel 1000 geeft aan hoeveel stringbytes het programma nodig heeft. Regel 20 reserveert de benodigde ruimte in computers waar dat nodig is, brengt de computer in de BASICODE-stand, stelt het scherm in de juiste kleur in en regelt nog het een en ander dat geregeld moet worden. Daarna wordt naar regel 1010 gesprongen voor het eigenlijke BASI-

CODE-programma. Als dat is afgelopen springt het naar regel 950 waar de computer weer in de normale werkstand, kleur e.d. wordt teruggebracht.

4.3 De BASICODE-3 standaardroutines

In deze paragraaf worden de standaard-subroutines beschreven, zoals die op de regels met nummers lager dan 1000 te vinden zijn. Deze subroutines zijn voor iedere computer apart geschreven, zodat ze er voor iedere BASIC anders uitzien. Ze doen echter in alle computers precies hetzelfde!

GOTO 2Ø (programma-aanloop)

Deze opdracht (eigenlijk dus nog geen subroutine!) mag en moet alleen in regel 1000 staan. Hiermee wordt de computer in de BASICODE-toestand gebracht, wordt het scherm gewist, wordt de benodigde hoeveelheid ruimte voor strings gereserveerd, worden alle variabelen gewist en worden de variabelen HO en VE voorzien van de maten van het tekstscherm en worden in HG en VG de maten van het grafische scherm gezet. Daarna wordt gesprongen naar regel 1010 in het programma. Regel 1010 is eigenlijk de eerste regel van het programma. De waarden die in HO en VE worden gezet, zijn beide één minder dan het aantal tekens per regel resp. het aantal tekstregels dat mogelijk is. In een computer met 40 tekens per regel zal dus in HO de waarde 39 komen te staan. In HG en VG wordt opgegeven hoeveel beeldpunten er in grafisch bedrijf in horizontale en verticale richting getoond kunnen worden. De betekenis hiervan zal bij de bespreking van de grafische subroutines duidelijk worden.

GOSUB 1ØØ (tekstbedrijf en scherm wissen)

Deze subroutine schakelt (indien nog nodig) om naar tekstbedrijf, wist het scherm en plaatst de cursor op de positie 0,0 (dat is linksboven op het scherm). Er wordt geen enkele variabele van waarde veranderd, dus ook niet HO en/of VE.

GOSUB 11Ø (cursor naar HO,VE)

De subroutine plaatst de cursor op positie HO,VE op het scherm. HO is de positie op de regel; daarbij komt de waarde 0 overeen met de meest linkse positie. VE is het nummer van de schermregel; de bovenste schermregel is regel 0.

In veel computers is het scherm 40 karakters lang en 24 karakters hoog. In die gevallen mag HO dus niet groter dan 39 en VE niet groter dan 23 zijn.

De variabelen HO en VE veranderen niet van waarde door aanroep van deze subroutine. Een slim programma maakt bij het bepalen van schermposities gebruik van de waarden die bij de start (na GOTO 2Ø) in HO en VE worden gezet.

GOSUB 12Ø (cursorpositie in HO,VE)

Geef de positie waar de cursor op het scherm staat en zet die in de variabelen HO en VE. Als dus na afloop HO=Ø dan staat de cursor op de meest linkse positie op een regel en als VE=Ø dan staat de cursor in de bovenste regel van het scherm. Samen met de vorige subroutine kunt u bijvoorbeeld de cursor één of meer regels hoger of lager zetten:

```
2ØØØ GOSUB 12Ø :REM VRAAG CURSORPOSITIE OP
2Ø1Ø HO=Ø:GOSUB 11Ø:REM PLAATS DE CURSOR VOORAAN
2Ø2Ø PRINT"                    ":REM WIS BEGIN VAN DE REGEL
2Ø3Ø GOSUB 11Ø :REM POSITIONEER OPNIEUW
2Ø4Ø PRINT"Dit staat links"
2Ø5Ø VE=VE+1:GOSUB 11Ø :REM POSITIONEER CURSOR
2Ø6Ø PRINT"en dit staat er onder"
```

GOSUB 15Ø (print opvallend)

Print de inhoud van de string SR$ op een opvallende wijze op het scherm. In veel computers wordt dit gerealiseerd door de tekst van SR$ in 'reverse-field' weer te geven. Voor en na de tekens van SR$ worden op het scherm drie spaties geprint; sommige daarvan kunnen eveneens op opvallende wijze worden weergegeven. In totaal worden dus zes tekens meer geprint dan met de lengte van SR$ overeenkomt. Het geheel mag niet 'overlopen' op de volgende schermregel. Een voorbeeld:

```
1Ø1Ø SB=HO :REM BEWAAR SCHERMBREEDTE
.
.
2Ø9Ø HO=INT(SB/2-1Ø):IF HO<Ø THEN HO=Ø
21ØØ VE=Ø:GOSUB 11Ø
22ØØ SR$="KOP VAN HET PROGRAMMA":GOSUB 15Ø
```

Na regel 2200 zal de tekst "KOP VAN HET PROGRAMMA" op opvallende wijze in het midden van de bovenste schermregel te lezen zijn. Waarschuwing: gebruik deze subroutine met mate! Als álles opvallend is dan valt er juist niets meer op!

GOSUB 2ØØ (kijk naar toetsenbord)

Kijk of er een toets is ingedrukt. Zo ja, zet de stringwaarde van deze toets in de variabele IN$. Zo nee, zet in IN$ een lege string. Tevens wordt in de variabele IN de "echte" ASCII-waarde van de (eventueel) ingedrukte toets afgegeven. Daarmee wordt bedoeld: een getal, overeenkomstig de tabel in appendix 4, dat is dus in normale gevallen minstens 32 en hoogstens 95. Ongeacht of een kleine letter of een hoofdletter is ingedrukt, voor bijvoorbeeld de "A" en de "a" wordt in beide gevallen de code 65 afgegeven. Hetzelfde geldt voor tekens als } en] welke beide in IN de code 94 geven. Cijfertoetsen geven codes van 48 tot en met 57. Als geen toets is ingedrukt, komt in IN de waarde 0 te staan.

In principe kunnen alle toetsaanslagen in IN$ komen te staan, dus ook controletekens zoals RETURN of een druk op een andere besturingstoets. Let echter op: de controletekens hebben bepaald niet bij alle computers dezelfde getalwaarde of stringwaarde, gebruik ze dus liever niet in een programma. Uitzonderingen daarop zijn de volgende:

De RETURN (of ENTER, NEW LINE enz.) toets heeft bij alle computers de code 13, dus bij het ingedrukt zijn van deze toets wordt IN=13 en IN$=CHR$(13).

Verder wordt bij het aanraken van bepaalde besturingstoetsen gegarandeerd de volgende code in IN (als getal) afgegeven (dus ook als de computer van nature bij die toets een andere code zou afgeven):

cursor links	: 28	(decimaal)
cursor rechts	: 29	
cursor omlaag	: 30	
cursor omhoog	: 31	
wis/delete	: 127	

N.B.1: Na afloop van deze routine kan, afhankelijk van het type computer, ASC(IN$) verschillen van de waarde in IN en eveneens kan CHR$(IN) verschillen van IN$. Dit mogelijke verschil kan in principe optreden voor ALLE karakters.

N.B.2: Indien een cursor- of deletetoets was ingedrukt, is de waarde in IN$ in feite onbepaald. Testen op IN$ of een opdracht als PRINT IN$ is dan niet zinvol.

Een voorbeeldje:

```
3ØØØ GOSUB 11Ø :REM POSITIONEER CURSOR
3Ø1Ø GOSUB 2ØØ :REM TEST TOETSENBORD
3Ø2Ø IF IN<>127 THEN 3Ø5Ø
3Ø3Ø HO=HO-1:GOSUB 11Ø
3Ø4Ø PRINT" ";:GOTO 3ØØØ
3Ø5Ø IF IN=13 THEN HO=Ø:VE=VE+1
3Ø6Ø IF IN=28 THEN HO=HO-1
3Ø7Ø IF IN=29 THEN HO=HO+1
3Ø8Ø IF IN=3Ø THEN VE=VE+1
3Ø9Ø IF IN=31 THEN VE=VE-1
3Ø95 IF HO<Ø THEN HO=Ø:GOSUB25Ø
31ØØ IF IN>31 THEN PRINT IN$;:HO=HO+1
311Ø IF HO>SB THEN HO=Ø:VE=VE+1
3115 IF VE<Ø THEN VE=SH
312Ø IF VE>SH THEN VE=SH:GOSUB 25Ø:REM PIEP
313Ø GOTO....
```

GOSUB 21Ø (wacht op toetsindruk)

Wacht tot er een toets is ingedrukt, zet de stringwaarde van deze toets in de variabele IN$ en de code in de variabele IN. Het verschil met de vorige routine is dat deze subroutine blijft wachten tot een toets wordt ingedrukt, terwijl de vorige alleen even keek of er op dat moment toevallig een toets ingedrukt was.

Ook bij deze subroutine wordt op dezelfde wijze in de variabele IN de echte ASCII-waarde afgegeven. Zie daarvoor de hierboven bij subroutine 200 gegeven nadere uitleg.

GOSUB 22Ø (lees van het tekstscherm)

Deze subroutine levert in IN de code van het karakter dat op positie HO,VE op het scherm zichtbaar is. Indien HO,VE buiten het scherm valt, levert de subroutine als resultaat IN=Ø af. In het andere geval voldoet de waarde in IN aan de bij de subroutine 200 vermelde regel dat de waarde in IN altijd ten minste 32 is en ten hoogste 95 (behoudens ingedrukte besturingstoets). Als dus op het scherm een kleine letter "a" zichtbaar is, dan zal in IN de waarde 65 (van "A") worden afgeleverd. De waarde van IN$ wordt door deze subroutine niet veranderd.

GOSUB 25Ø (attentiesein)

Deze subroutine produceert een piepje, dit om een attentiesignaal aan de gebruiker te geven. De toonhoogte en -duur zijn niet exact vastgelegd, zodat de routine niet geschikt is voor het maken van muziek. Gebruik deze subroutine verstandig. Veel gebruikers van programma's worden door te veel piepjes eerder kwaad, dan dat ze extra aandachtig worden.

GOSUB 26Ø (geef randomgetal)

Deze subroutine geeft een willekeurig getal in de variabele RV. Dit getal is altijd kleiner dan 1 en ten minste 0. Zoiets is nuttig in statistische programma's en bij spelletjes.

GOSUB 27Ø (meld vrije ruimte)

Deze subroutine schoont de variabelenruimte op en bepaalt hoeveel geheugenruimte er nog vrij is. (De variabelen worden niet gewist!) Het aantal nog vrije bytes komt in de variabele FR te staan. Programma's die veel geheugenruimte nodig hebben, kunnen via deze subroutine te weten komen of de benodigde ruimte beschikbaar is. Bijvoorbeeld:

```
4ØØØ GOSUB 27Ø
4Ø1Ø L=INT(SRQ((FR-5ØØ)/6))
4Ø2Ø DIM MA(L,L)
```

Aangenomen dat voor elk getal zes geheugenplaatsen gebruikt worden, dimensioneert men op deze manier een zo groot mogelijke vierkante matrix, waarbij voor andere doeleinden nog minstens 500 bytes geheugenruimte gereserveerd blijven.

GOSUB 28Ø (STOP-toets schakelen)

Als bij de aanroep van deze subroutine de waarde in FR gelijk aan 1 is, dan wordt de stoptoets (breaktoets, escapetoets e.d.) uitgeschakeld. Als FR=Ø dan wordt deze toets juist weer in de normale werktoestand gebracht. Dit is speciaal van belang voor programma's die in het onderwijs worden gebruikt en waar het niet de bedoeling is dat een leerling het programma kan stoppen.

GOSUB 3ØØ (omzetting in string)

Maak een string SR$ die de waarde aangeeft van de variabele SR. De string bevat geen spaties aan het begin of einde van het getal, dit in tegenstelling tot de BASIC-functie STR$( ) die dat bij sommige BASIC-dialecten juist wel doet. Het gebruik van de functie STR$( ) is in BASICODE-3 niet toegestaan.

Een voorbeeldje (let op de spaties in regel 5010):

```
5ØØØ SR=VR:GOSUB 3ØØ
5Ø1Ø PRINT"Je hebt nu ";SR$;" vragen gehad"
```

GOSUB 31Ø (getal formatteren)

Deze subroutine maakt in SR$ een string die bepaald wordt door de variabelen CT, CN en SR. SR$ zal net als in de vorige routine de waarde aangeven van SR. De totale lengte van SR$ bedraagt nu exact CT karakters, waarvan CN karakters na de decimale punt. Als het getal niet in het opgegeven formaat past, bestaat SR$ uit CT sterren. Zonodig wordt SR correct afgerond. De variabelen CT, CN en SR worden niet veranderd door aanroep van de routine. Enkele voorbeelden:

```
CT=7:CN=3:SR=2/3:GOSUB 31Ø          geeft SR$=" Ø.667"
CT=8:CN=5:SR=-1.1E-3:GOSUB 31Ø      geeft SR$="-Ø.ØØ11Ø"
CT=8:CN=2:SR=-1.1E-3:GOSUB 31Ø      geeft SR$="    Ø.ØØ"
CT=3:CN=Ø:SR=23.6:GOSUB 31Ø         geeft SR$=" 24"
CT=3:CN=1:SR=23.6:GOSUB 31Ø         geeft SR$="***"
```

GOSUB 33Ø (hoofdletters maken)

Deze subroutine verandert alle eventueel in SR$ aanwezige kleine letters in hoofdletters. Na afloop kan bijvoorbeeld MID$(SR$,3,1) wel "A" maar nooit "a" zijn. In

programma's waar gegevens worden gesorteerd op alfabetische volgorde kan dit erg handig zijn.

GOSUB 350 (tekst naar printer)

Druk SR$ af op de printer, maar sluit de regel nog niet af. U kunt dus nog meer op dezelfde regel printen via deze routine. Op andere wijze iets naar de printer sturen is in BASICODE-3 niet toegestaan.

Let op: lang niet iedereen heeft een printer ter beschikking! Vraag daarom in het programma aan de gebruiker om een keus te maken tussen afdrukken op de printer en afdrukken op het scherm.

GOSUB 360 (regelopvoer printer)

Sluit de regel op de printer af en begint op een nieuwe printerregel. Als op een nieuw blad papier moet worden begonnen, zal het nodig zijn dat het programma zelf bijhoudt hoeveel regels reeds zijn geprint en op grond daarvan het juiste aantal malen deze subroutine aanroept.

GOSUB 400 (muziek)

Deze subroutine produceert een toon. De gewenste toon wordt opgegeven in drie variabelen:

SP (Sound Pitch) voor de toonhoogte,	0 <= SP <= 127
SD (Sound Duration) voor de duur,	1 <= SD <= 255
SV (Sound Volume) voor de sterkte.	0 <= SV <= 15

De subroutine wordt pas beëindigd als de toon is afgelopen.
De variabelen worden als volgt gedefinieerd:

SV = 0	absolute stilte
SV = 7	gemiddelde, normale sterkte
SV = 15	maximale sterkte

SD	de tijdsduur in eenheden van 100 milliseconden. SD=10 komt dus overeen met een duur van 1 seconde.

SP = 1	de laagste toon, een akelig gebrom
SP = 60	de 'centrale C'
SP = 69	de 'standaard' A, meestal circa 440 Hz
SP = 127	de hoogste toon, voor veel mensen onhoorbaar

Voor elke halve toon hoger of lager is SP 1 hoger resp. lager. Elk octaaf omvat 12 tonen. Het is niet zinvol alle mogelijke waarden van SP te gebruiken. Ten eerste omdat

de erg lage en erg hoge waarden geen aangenaam geluid opleveren en ten tweede omdat niet elke computer in staat is het volle bereik van 127 tonen te laten horen. Voor dit laatste zij ook verwezen naar appendix 1.

GOSUB 45Ø (afbreekbare wachtroutine)

Bij het aanroepen van deze subroutine moet in SD een getal staan dat een wachttijd in eenheden van 100 milliseconden aangeeft. SD=1Ø komt dus overeen met precies 1 seconde. De subroutine wordt automatisch verlaten als de opgegeven tijd om is. In dat geval is bij het verlaten van de routine SD=Ø, IN$="" en IN=Ø.

Als echter gedurende de wachttijd een toets wordt ingedrukt, wordt de routine onmiddellijk beëindigd; in SD staat dan de nog resterende tijd in eenheden van 100 milliseconden, in IN$ de ingedrukte toets en in IN de code van die toets. Zie voor de codes in IN de opmerkingen bij subroutine 200.

Opmerking: Voor een niet-afbreekbare wachtroutine kunt u gebruik maken van GOSUB 4ØØ met SV=Ø.

Een voorbeeld:

```
6ØØØ PRINT"Druk op een toets...."
6Ø1Ø SD=5Ø:GOSUB 45Ø
6Ø2Ø IF SD>Ø THEN 6Ø5Ø
6Ø3Ø PRINT"Na 5 seconden heb je nog steeds niets ingedrukt!"
6Ø4Ø GOTO 6Ø8Ø
6Ø5Ø SR=SD/1Ø:CN=1:CT=4:GOSUB 31Ø
6Ø6Ø PRINT"Het duurde ";SR$;" seconden"
6Ø7Ø PRINT"voor je de ";IN$;" indrukte."
6Ø8Ø RETURN
```

GOSUB 5ØØ (bestand openen)

Bij het aanroepen van deze subroutine moet in NF$ de naam van het te openen bestand staan (max. 7 tekens) en in NF een code:
NF=Ø : openen voor invoer vanaf BASICODE-cassette
NF=1 : openen voor uitvoer naar BASICODE-cassette
NF=2 : openen voor invoer vanaf eigen achtergrondgeheugen
NF=3 : openen voor uitvoer naar eigen achtergrondgeheugen
NF=4 : openen voor invoer vanaf diskette
NF=5 : openen voor uitvoer naar diskette
NF=6 : openen voor invoer vanaf diskette
NF=7 : openen voor uitvoer naar diskette

Kortom:
even nummers = invoer
oneven nummers = uitvoer
nummers 0 en 1 : cassette in BASICODE-formaat (zie verder) (van NF$ wordt geen gebruik gemaakt);
nummers 2 en 3 : diskette of cassette op de manier waarop de computer het normaal zelf doet;
nummers 4 en 5 : een tweede bestand, hetgeen alleen mogelijk is naar diskette;
nummers 6 en 7 : een derde bestand, wederom alleen op diskette.

Cassettes, beschreven met bestanden in BASICODE-formaat (NF=1), kunnen door alle BASICODE-computers worden ingelezen (NF=Ø) en verwerkt. Op deze wijze kunnen computers van verschillende merken dus gegevens uitwisselen.

Tijdens het werken van deze en de volgende bestandsroutines kunnen op de onderste schermregel bepaalde meldingen komen. (Bijvoorbeeld: 'Zet de recorder op weergave'.) Deze meldingen dienen natuurlijk door de gebruiker te worden opgevolgd. Tijdens het lezen of schrijven van cassette zal op het scherm (onderste regel) enige activiteit zichtbaar zijn. Dit dient als controle op de werking van de routines. Na afloop van de routine worden deze meldingen weer gewist. Daarbij kan dus de onderste schermregel worden gewist.

Opmerking: Naar diskette kunnen dus maximaal drie bestanden tegelijk geopend zijn. Er is geen garantie (integendeel) dat elke BASICODE-computer daartoe in staat is.

GOSUB 54Ø (invoer uit bestand)

Bij aanroep van deze subroutine staat in NF de bestandscode. De routine levert in IN$ de eerstvolgende string vanuit het tevoren geopende bestand en tegelijk in IN de waarde 0. Als echter in IN$ de laatste string uit het bestand wordt afgeleverd, wordt in IN de waarde 1 afgegeven. Als de subroutine wordt aangeroepen nadat de laatste string reeds is afgeleverd dan wordt IN$="" en IN=1. Als de subroutine een fout ontdekt, bijvoorbeeld een leesfout vanaf de cassette, dan wordt IN=-1 gemaakt. Indien geen antwoord in IN$ mogelijk is, bijvoorbeeld als het bestand niet geopend is, wordt IN$="" en IN=-1 afgeleverd.

GOSUB 56Ø (uitvoer naar bestand)

De inhoud van SR$ wordt naar het bestand met codenummer NF geschreven. In de variabele IN worden eventuele fouten gemeld: IN=Ø als alles goed is, IN=-1 als de gevraagde uitvoer onmogelijk is.

GOSUB 58Ø (bestand afsluiten)

Het bestand met code NF wordt afgesloten. In de variabele IN worden eventuele fou-

ten gemeld: IN=∅ als alles goed is, IN=-1 als de gevraagde actie onmogelijk is.

Een programmavoorbeeld voor de bestandsroutines:

```
8∅∅∅ NF=1:GOSUB 5∅∅ :REM OPEN VOOR UITVOER
8∅1∅ FOR I=1 TO 1∅
8∅2∅ PRINT"Tik een naam in ";
8∅3∅ INPUT SR$
8∅4∅ GOSUB 56∅ :REM NF=1
8∅6∅ NEXT I
8∅7∅ GOSUB 58∅ :REM BESTAND AFSLUITEN
8∅9∅ PRINT"Spoel de cassette terug en"
81∅∅ PRINT"druk dan op een toets"
811∅ GOSUB 21∅
812∅ NF=∅:GOSUB 5∅∅ :REM OPEN VOOR INVOER
813∅ PRINT"De namen waren:"
814∅ GOSUB 54∅:PRINT IN$
815∅ IF IN=∅ THEN 814∅
816∅ GOSUB 58∅
817∅ ....
```

GOSUB 6∅∅ (grafisch bedrijf en scherm wissen)

Deze subroutine schakelt om naar grafisch bedrijf en wist het scherm. De grafische cursor wordt in het punt 0,0 geplaatst, hetgeen overeenkomt met de linkerbovenhoek van het grafische scherm. De indeling van het grafische scherm is aldus:

0,0 is het punt precies linksboven op het grafische scherm,
1,0 is het punt precies buiten de rechterbovenhoek,
0,1 is het punt precies onder de linkeronderhoek en
1,1 ligt schuin onder de rechteronderhoek van het grafische scherm.

N.B.1: In grafisch bedrijf hebben de gewone PRINT-opdracht en de subroutines 100, 110, 120, 150 en 220 geen betekenis. Deze mogen dus niet gebruikt worden alvorens met GOSUB 1∅∅ weer is teruggeschakeld naar tekstbedrijf.

N.B.2: Gemeten op het beeldscherm is de afstand van het punt (0,0) naar het punt (0,1) exact 3/4 (75%) van de afstand van (0,0) naar (1,0). Het punt (1/2,1/2) komt in het midden van het beeldscherm.

GOSUB 62∅ (plot een punt)

Deze subroutine zet de grafische cursor op het punt HO,VE en plot dat punt. Zowel HO als VE moeten minstens 0 zijn en kleiner dan 1. Als CN=∅ wordt het schermpunt aangezet (voorgrondkleur), als CN=1 wordt dat punt juist uitgezet (achtergrondkleur).

GOSUB 63Ø (trek een lijnstuk)

Deze subroutine doet bijna hetzelfde als subroutine 620; echter, nu wordt niet alleen het opgegeven punt geplot maar een compleet lijnstuk vanaf waar de grafische cursor stond naar het in HO en VE opgegeven punt. Het gehele lijnstuk wordt in de met CN aangegeven kleur getrokken. Na afloop staat de grafische cursor in het door HO en VE aangegeven eindpunt.

GOSUB 65Ø (tekst op het grafische scherm)

Deze subroutine print de tekst in SR$ op het grafische scherm. HO en VE moeten de (grafische) positie aangeven van de linkerbovenhoek van het eerste karakter. De karakters worden zo veel mogelijk op het scherm afgebeeld in het 'normale' formaat, waarbij er dus in principe 40 op het scherm passen. Dit wordt echter niet in alle computers gegarandeerd, zie appendix 1. Aanbevolen wordt daarom in SR$ niet te veel karakters op te geven. De tekst wordt weergegeven in voorgrondkleur (als CN=Ø) of in achtergrondkleur (als CN=1). Na afloop staat de grafische cursor in HO,VE.
Een eenvoudig voorbeeld:

```
9ØØØ GOSUB 6ØØ :REM GRAFISCH BEDRIJF
9Ø1Ø CN=Ø:HO=Ø.2:VE=Ø.1:GOSUB 62Ø
9Ø2Ø HO=Ø.8:GOSUB 63Ø
9Ø3Ø VE=Ø.9:GOSUB 63Ø
9Ø4Ø HO=Ø.1:GOSUB 63Ø
9Ø5Ø VE=Ø.1:GOSUB 63Ø
9Ø6Ø SR$="LINKS ONDER":GOSUB65Ø
```

Dit stukje programma tekent op het scherm een groot vierkant met tegen de linkeronderhoek aan de woorden "LINKS ONDER".

GOTO 95Ø (Programma-beëindiging)

De opdrachten END en STOP zijn in BASICODE-3 niet toegestaan. In plaats daarvan moet men gebruiken: GOTO 95Ø. Daar wordt de machine weer in de toestand gebracht waarin hij normaal ook na inschakelen staat, dus cursor aan, stoptoets werkend, normaal scherm, normale stringruimte enz. Daarna wordt het scherm gewist en eindigt het programma. Echter: het programma wordt niet aangetast en de BASICODE-routines worden niet vernield.

GOTO 1ØØØ (Programmastart)

De instructie RUN is in een BASICODE-3 programma verboden. Als het nodig is, vanuit het programma opnieuw te beginnen (bijvoorbeeld om arrays te wissen en opnieuw te dimensioneren) dan moet men (verplicht) gebruiken: GOTO 1ØØØ.

4.4 Overzicht BASICODE-3 subroutines

10	eerste regel van het subroutineblok
20	programmastart, systeemreset, variabelen wissen, enz. ..
100	schakel om naar tekstbedrijf en wis het scherm
110	verplaats de cursor naar positie HO,VE
120	registreer de cursorpositie in HO,VE
150	print op opvallende wijze 3 spaties; SR$; 3 spaties
200	geef een eventueel ingedrukte toets in IN$ en IN
210	wacht tot toetsindruk en geef deze in IN$ en IN
220	geef in IN de code van wat op schermpositie HO,VE te zien is
250	geef een piepje als attentiesein
260	geef random in RV zodat 0 <= RV < 1
270	doe 'garbage collect' en geef in FR het totaal aantal vrije bytes
280	schakel de stoptoets in (FR=Ø) of uit (FR=1)
300	geef in SR$ de tekstvorm van SR, zonder spaties
310	geef in SR$ de tekstvorm van SR, geformatteerd conform CT en CN
330	verander alle kleine letters in SR$ in hoofdletters
350	stuur SR$ naar de printer
360	sluit de printerregel af met wagenterug en papieropvoer
400	maak een toon volgens SV, SD en SP
450	wacht maximaal SD*100 milliseconden op een toetsindruk
500	open bestand NF$ volgens code NF
540	lees in IN$ uit geopend bestand NF en in IN de status
560	voer SR$ uit naar het geopende bestand met code NF
580	sluit het bestand met code NF af
600	schakel om naar grafisch bedrijf en wis het scherm
620	plot het punt op positie HO,VE in kleur CN
630	trek een lijn naar punt HO,VE in kleur CN
650	print SR$ als tekst vanaf positie HO,VE
950	beëindig het programma en schakel de machine naar normaal bedrijf

4.5 Variabelen en BASIC-opdrachten

De variabelen die in het programma gebruikt worden, zijn aan enkele beperkingen gebonden. Dit is nodig om de uitwisselbaarheid van programma's te waarborgen.

a. Numerieke variabelen zijn 'real' en in enkele precisie. De nauwkeurigheid is, afhankelijk van de computer, soms niet meer dan 6 decimalen; reken in elk geval nooit op een grotere nauwkeurigheid.

b. De namen van de variabelen zijn maximaal twee tekens lang, waarbij het eerste teken een hoofdletter moet zijn. Indien er nog een tweede teken volgt, mag dat een hoofdletter of een cijfer zijn. In namen van variabelen mogen dus geen kleine letters gebruikt worden. Voor stringvariabelen wordt de naam gevolgd door $. Alle andere toevoegingen (zoals bijvoorbeeld %, # en !) zijn in BASICODE-3 verboden.

c. Logische variabelen zijn variabelen die 'waar' of 'onwaar' zijn. Er mag geen gebruik gemaakt worden van de eventuele numerieke waarde van logische varia-

belen. In sommige computers wordt 'waar' namelijk voorgesteld door +1, in andere door −1. Het resultaat kan alleen gebruikt worden in een IF . . . THEN-constructie. (iets als A=3*(B=1) mag dus absoluut niet! Gebruik in zo'n geval A=Ø:IF B=1 THEN A=3).

d. Voordat een variabele gebruikt wordt, moet hij altijd een waarde krijgen. U mag dus niet aannemen dat de variabelen automatisch op nul gezet worden bij het starten van het programma.

e. Stringvariabelen mogen maximaal 255 karakters lang zijn.

f. Namen van variabelen mogen niet beginnen met de letter O, zulke namen zijn voorbehouden aan gebruik binnen de BASICODE-3 standaardroutines.

g. Uitgesloten zijn tevens de variabelen: AS, AT, FN, GR, IF, LN, PI, ST, TI, TI$, TO.

h. Bij het aanroepen van de BASICODE-3 subroutines kan gebruik gemaakt moeten worden van de variabelen: CN, CT, FR, HG, HO, IN, IN$, NF, NF$, RV, SD, SP, SR, SR$, SV, VE, VG.

4.6 Toegestane BASIC-opdrachten

Deze paragraaf geeft geen uitputtende verhandeling over programmeren in BASIC. Van elke opdracht wordt een korte, zakelijke beschrijving gegeven omdat het BASICODE-3 protocol enige beperkingen oplegt aan het gebruik van sommige opdrachten.

Uitsluitend de volgende BASIC-opdrachten en operatoren mogen in de programma's gebruikt worden:

ABS	AND	ASC	ATN	CHR$	COS	DATA	DIM
EXP	FOR	GOSUB	GOTO	IF	INPUT	INT	LEFT$
LEN	LET	LOG	MID$	NEXT	NOT	ON	OR
PRINT	READ	REM	RESTORE	RETURN	RIGHT$	SGN	SIN
SQR	STEP	TAB	TAN	THEN	TO	VAL	

^	*	/	+	−	
=	<	>	<=	>=	<>

Hierna vindt u in alfabetische volgorde de beschrijving van de werking van de toegestane BASIC-opdrachten.

ABS Geeft de absolute waarde van de meegegeven variabele.
Voorbeelden:

```
A= 1Ø :B=ABS(A)    na afloop is B=10
A=-2Ø :B=ABS(A)    na afloop is B=20
A=-1  :B=ABS(A-5)  na afloop is B=6
```

AND

Logische AND. Mag alleen worden gebruikt voor logische variabelen. Het resultaat is een logische uitkomst. Het gebruik van haakjes om de bewerkingsvolgorde duidelijk aan te geven, is in BASICODE-3 verplicht. Voorbeelden:

```
IF (A=5) AND (B=Ø) THEN ...
Q=(A=5) AND (B=Ø): IF Q THEN ...
```

ASC

Geeft de ASCII-waarde van het eerste karakter van de meegegeven string. Voorbeelden:

```
A$="A":B=ASC(A$)     meestal geldt nu: B=65
A$="BEER":B=ASC(A$)  meestal geldt nu: B=66
```

N.B.: Bij sommige merken computers wordt intern een andere code gebruikt voor de diverse karakters. Het kan dan bijvoorbeeld gebeuren dat in bovenstaande voorbeelden in B de getallen 193 resp. 194 komen te staan. Waar het letters en/of cijfers betreft, geldt wel steeds dat de afwijkende waarden een veelvoud van 32 verschillen van de 'normale' waarde. Het gebruik van de ASC-functie dient dus met de nodige omzichtigheid te geschieden en kan het best zo veel mogelijk vermeden worden.

ATN

Geeft de arctangens in radialen van de meegegeven variabele. Voorbeeld:

```
PRINT ATN(1)     geeft als resultaat .785398
PRINT ATN(-1.5)  geeft als resultaat -0.982794
```

CHR$

Geeft een karakter dat in de computer overeenkomt met de waarde van de meegegeven variabele. De variabele mag lopen van 32 t/m 127. Bij waarden kleiner dan 32 is voorzichtigheid geboden daar de controlkarakters bij diverse computers sterk verschillen, alleen de RETURN-toets heeft altijd ASCII-code 13. Bovendien kennen niet alle computers kleine letters, dus de ASCII-codes groter dan 96 voorzichtig gebruiken. Voorbeeld:

```
A$=CHR$(66)  A$ bevat nu (meestal) de letter B.
```

Zoals reeds bij ASC vermeld, zijn er computers die intern afwijkende codes hanteren. Gebruik daarom ook de CHR$-functie alleen met de grootst mogelijke voorzichtigheid.

COS — Geeft de cosinus van de meegegeven hoek. De variabele dient de hoek in radialen te bevatten. Voorbeeld:

`PRINT COS(1)` geeft als resultaat .540302

DATA — Na dit commando volgen tot het einde van de regel waarden voor variabelen die met READ gelezen kunnen worden. Op de DATA-regel mogen na de met READ te lezen variabelen geen andere commando's (bijvoorbeeld REM) voorkomen. De variabelen worden gescheiden door een komma. Stringvariabelen moeten tussen aanhalingstekens staan. Voorbeeld:

```
DATA 1ØØ,2ØØ,"HALLO","BASICODE",4.6,89
```

DIM — Moet altijd gebruikt worden om arrays te dimensioneren. Een array mag slechts éénmaal in een programma gedimensioneerd worden, en wel voordat het gebruikt wordt in het programma. Het aantal dimensies is in BASICODE-3 maximaal twee, het aantal elementen wordt beperkt door de grootte van het geheugen. Verschillende arrays mogen met dezelfde DIM-opdracht worden gedimensioneerd, zie het voorbeeld.

N.B.1: Bij sommige computers behoeven arrays tot 10 elementen niet te worden gedimensioneerd. In BASICODE-programma's moet dit per se altijd!

N.B.2: Ook het element met nummer 0 doet mee, dus A(Ø) en AD$(Ø,Ø) bestaan ook.

Voorbeeld: `DIM A$(12),HD(1ØØ,1ØØ),MP(1ØØØ)`

EXP — Verheft het getal e (2.71828...) tot een bepaalde macht. Voorbeeld:

`PRINT EXP(2)` geeft als resultaat 7.38906

FOR — FOR ... TO ... STEP ... NEXT ... Lusconstructie; de lus wordt minstens éénmaal doorlopen. STEP en de waarde daarna mag worden weggelaten, de stapgrootte is dan 1. Na NEXT moet altijd de variabele worden meegegeven! Meer dan één variabele na NEXT is niet toegestaan. Voorbeelden:

```
FOR X=1Ø TO 1ØØ
PRINT X,X+34
NEXT X
```

```
FOR C=A TO B STEP -3
PRINT C,C-B
NEXT C

FOR I=1 TO 1Ø: FOR J=1 TO 5
PRINT I;"*";J;"=";I*J
NEXT J: NEXT I
```

N.B.: Voorkom springen uit een FOR–NEXT-lus zonder dat deze is afgemaakt! Verlaat een FOR–NEXT-lus dus alleen via NEXT, desnoods door de lusvariabele op de eindwaarde te zetten.
Gebruik in gevallen waar u uit de lus zou willen wegspringen geen FOR–NEXT voor de lusstructuur.

GOSUB — Roep een subroutine aan, beginnend op het regelnummer dat achter dit commando staat vermeld. Voorbeeld:

```
GOSUB 1ØØ
```

N.B.: Verboden is: A=1ØØ:GOSUB A

GOTO — Spring naar het regelnummer dat achter dit commando staat vermeld. Voorbeeld:

```
GOTO 15ØØ
```

N.B.1: Verboden is: A=15ØØ:GOTO A

N.B.2: Een GOTO-opdracht mag nimmer springen naar een niet bestaand regelnummer of naar een regelnummer in de BASICODE-subroutines. Op dit laatste zijn twee uitzonderingen:
-1- In regel 1010 moet GOTO2Ø staan en
-2- GOTO95Ø dient ten minste eenmaal in het programma voor te komen als beëindiging van het programma (in plaats van de in de meeste BASIC-versies gebruikelijke END- of STOP-opdracht die in BASICODE niet is toegestaan).

IF — IF . . . THEN Voorwaardelijke splitsing. Tussen IF en THEN staat een logische variabele of logische vergelijking. Als deze 'waar' is, gaat de uitvoering door na THEN; zoniet dan gaat de uitvoering verder op de volgende BASIC-regel. Ook kan na THEN een regelnummer volgen waar de uitvoering van het programma moet doorgaan.

N.B.: ELSE is niet toegestaan in BASICODE-3.

Voorbeelden:

```
IF A=3 THEN B=Ø:C=5
IF A>3 THEN 15ØØ
IF A>3 THEN GOTO 15ØØ
C=(A>3):IF C THEN GOSUB 1ØØ
```

Let op:
NIET: IF ... GOTO 2ØØØ maar IF ... THEN 2ØØØ
NIET: IF ... GOSUB 2ØØØ maar IF ... THEN GOSUB 2ØØØ

INPUT

Vraagt aan de gebruiker om invoer die wordt toegekend aan de variabele volgend op INPUT. Deze variabele mag een numerieke variabele of een stringvariabele zijn. Een ingevoerde string mag geen komma's of 'dubbele punten' bevatten. Wanneer dat wél gewenst is, kunt u beter gebruik maken van de subroutine op regel 210. Een "promptstring" is niet toegestaan, evenmin als meer dan één variabele na een INPUT. De meeste BASIC's printen een vraagteken als deze opdracht wordt gegeven. Na het intikken van de returntoets wordt bij een aantal computers de rest van de regel vanaf de cursorpositie tot het einde gewist. Voorbeelden:

```
PRINT"Wat is Uw naam";:INPUT N$
PRINT"Tik de waarden in ";:INPUT A: INPUT B
```

N.B.: INPUT"Uw naam";A$ is dus niet toegestaan!

INT

Geeft het grootste gehele getal dat hoogstens gelijk is aan de meegegeven variabele. Voorbeelden:

A=2.1:B=INT(A) na afloop is B=2
B=INT(-2.5) na afloop is B=-3

LEFT$

Geeft een aantal karakters van de meegegeven string te beginnen bij het meest linkse karakter. Het aantal karakters dat gevraagd wordt, moet minimaal 1 zijn en maximaal de lengte van de string.
Voorbeelden:

A$= LEFT$("BASICODE",5) na afloop is A$="BASIC"

N.B.: C$= LEFT$("BASICODE",Ø) mag dus niet!

LEN

Geeft de lengte van de meegegeven string. Voorbeelden:

A$="BASICODE-3":A= LEN(A$) na afloop is A=10
A$="":A=LEN(A$) na afloop is A=0

LET Mag gebruikt worden bij het toekennen van een waarde aan een variabele, is echter niet nodig. Voorbeeld:

`LET A=5` is hetzelfde als `A=5`

LOG Berekent de natuurlijke logaritme van de meegegeven variabele of uitdrukking. Voorbeelden:

`PRINT LOG(1)` geeft als resultaat 0
`PRINT LOG(1Ø)` geeft als resultaat 2.302585

MID$ Haalt een aantal karakters uit een string. `MID$(A$,X,Y)` geeft Y karakters van A$, te beginnen met het X-de karakter. (Het eerste karakter heeft nummer 1, X=0 of Y=0 is niet toegestaan.) Voorbeelden:

`A$="BASICODE IS LEUK":B$=MID$(A$,1Ø,2)`
B$ bevat nu "IS"

N.B.: Niet toegestaan is de opdracht `A$=MID$(C$,5)`

NEXT Afsluitende opdracht van een herhalingslus. (Zie ook FOR). Na NEXT moet altijd de bijbehorende variabele staan.

Voorbeelden: zie FOR . . .

NOT Logische ontkenning, alleen toepasbaar op logische variabelen. (Zie ook AND). Voorbeelden:

`A=5:B= NOT(A=6)` na afloop is B='waar'
`A=(5=5):B= NOT A` na afloop is B='onwaar'

ON ON . . . GOSUB . . . of
ON . . . GOTO . . .
Maakt een sprong naar een subroutine of een sprong naar een programmaregel. Na ON volgt een uitdrukking of variabele, na GOSUB of GOTO een reeks regelnummers onderling gescheiden door komma's. De waarde van de variabele of uitdrukking dient een geheel getal te zijn en bepaalt welk regelnummer gekozen wordt. Daartoe kunt u de regelnummers genummerd denken: als de variabele 1 is, wordt het eerste regelnummer gekozen, als de variabele 2 is het tweede regelnummer enz. De variabele mag niet groter kunnen worden dan het aantal opgegeven regelnummers.
Voorbeelden:

```
ON K GOTO 1100,3400,1500          K moet nu 1, 2 of 3 zijn!
ON K-5 GOSUB 6000,7000,3000       K moet nu 6, 7 of 8 zijn!
```

OR — Logische OR, mag slechts gebruikt worden bij logische variabelen. (zie ook AND). Voorbeelden:

```
IF (A=5) OR (B<3) THEN ...
C=(A=5) OR (B<3):IF C THEN ...
```

N.B.: Evenals bij AND is het gebruik van haakjes verplicht.

PRINT — Drukt een variabele of string af op het scherm. Verschillende variabelen in een PRINT-opdracht moeten gescheiden worden door een puntkomma. Indien geen automatische overgang naar de volgende regel gewenst is, moet aan het einde van de opdracht een puntkomma staan. Bij sommige computers worden bij het printen van een getal één of meer spaties voor en/of na de cijfers geprint. Als u dit niet wilt, kunt u de subroutines op 300 of 310 gebruiken. Voorbeelden:

```
A=5:A$="HALLO":PRINT A;" ";A$
5 HALLO
PRINT "HALLO";:PRINT " DAAR"
HALLO DAAR
CN=3:CT=5:SR=5:GOSUB310:PRINT "VIJF=";SR$
VIJF=5.000
```

READ — Leest de gegevens in de DATA-opdrachten en kent die toe aan de variabele(n) volgend op READ. Meer variabelen na een READ dienen gescheiden te worden door een komma. Na het starten van een programma begint het lezen bij de DATA-opdracht met het laagste regelnummer tot alle gegevens op die regel gelezen zijn. Daarna is de volgende DATA-regel aan de beurt. Let op: een numerieke variabele moet getallen lezen, een stringvariabele strings!
Voorbeelden:

```
DATA 1,"COMPUTER",3
READ A:READ A$:READ B    of
READ A,A$:READ B         of
READ A,A$,B
```

REM — Met deze opdracht kunt u commentaar in een programma zetten, om het voor anderen begrijpelijker te maken. Alles wat na REM staat tot het einde van de regel wordt door BASIC overgeslagen. Er mag in de regel geen dubbele punt voorkomen, dit geeft soms problemen.

RESTORE Begint het lezen met READ weer vanaf de eerste DATA-regel in het programma. Let op: na RESTORE mag geen regelnummer staan!

RETURN Geeft het einde van een subroutine aan. Na deze opdracht gaat de uitvoering van het programma verder bij de eerste opdracht volgend op de bijbehorende GOSUB. Een subroutine dient altijd met RETURN te worden afgesloten!

RIGHT$ Geeft een aantal karakters van een bepaalde string, eindigend bij het laatste karakter. Het minimumaantal gevraagde karakters moet 1 zijn, het maximumaantal de lengte van de string.
Voorbeelden:

A$="BASICODE":B$=RIGHT$(A$,4) na afloop is B$="CODE"

Let op: A$="PROTOCOL":A=Ø:B$=RIGHT$(A$,A)
mag niet omdat A=0

SIN Bepaalt de sinus van een in radialen opgegeven hoek. Zie verder bij COS.

SGN Geeft −1 als de variabele (of uitdrukking) negatief is, 0 als de variabele nul is en +1 als de variabele positief is.
Voorbeelden:

A=5:B=SGN(A) na afloop is B=1
A=-.ØØ1:B=SGN(A) na afloop is B =−1

SQR Bepaalt de wortel van een variabele of uitdrukking die niet negatief mag zijn. Voorbeeld:

A=SQR(2*32) na afloop is A=8

STEP Bepaalt de stapgrootte in een herhalingslus. Zie FOR ...

TAB Wordt in PRINT-opdrachten gebruikt om de cursor op een bepaalde plaats op de regel te zetten. De cursor kan alleen verderop in de regel gezet worden en afhankelijk van de computer worden spaties geprint of er blijft staan wat er al op de regel stond. TAB(Ø) is niet toegestaan. De meeste computers beginnen te tellen bij 0, er zijn echter ook computers die beginnen te tellen bij 1, daarom is het beter om de subroutine op regel 110 te gebruiken!

Voorbeeld:

```
PRINT"A";TAB(5);"B";TAB(1Ø);"C"
A    B    C
```
maar bij een andere computer:
```
A  B  C
```

TAN Berekent de tangens van een in radialen gegeven hoek. Zie verder bij COS

THEN Zie IF

TO Zie FOR

VAL Bepaalt de numerieke waarde van een string. Als de string niet zuiver numeriek is, is de uitkomst niet bij iedere computer hetzelfde. Voorbeeld:

`A$="1.4E6":A=VAL(A$)`	na afloop is A=1.4E6
`A$="12D":A=VAL(A$)`	na afloop is A onbepaald.

4.7 De toegestane operatoren

4.7.1 De rekenoperatoren

Op de rekenoperatoren is van toepassing: eerst machtsverheffen, dan van links naar rechts vermenigvuldigen en delen, dan van links naar rechts optellen en aftrekken. Daarenboven geldt: wat tussen haakjes staat, wordt eerst berekend en waarden van functies worden bepaald voordat het rekenen met de rekenoperatoren begint.

= Kent de waarde van de uitdrukking rechts van het = teken toe aan de variabele links van het = teken. Voorbeelden:

`A=4*6`	na afloop is A=24

^ Verheft een getal of variabele tot een bepaalde macht. Voorbeeld:

`A=2:B=16:C=A^B`	na afloop is C=65536

* Vermenigvuldigt twee getallen of variabelen met elkaar. Voorbeeld:

`A=5:B=3*2*A`	na afloop is B=30

/ Deelt twee getallen of variabelen op elkaar. Voorbeeld:

```
A=5:B=1ØØ/A/2           na afloop is  B=10
```

\+ Telt twee getallen of variabelen bij elkaar op. Voorbeeld:

```
B=1:A=B+9               na afloop is  B=10
```

– Trekt twee getallen of variabelen van elkaar af. Voorbeeld:

```
A=1Ø–3–4                na afloop is  A=3
```

4.7.2 De stringoperatoren

\+ Koppelt strings aan elkaar. Voorbeelden:

```
A$="BAS":B$="ICO":C$="DE-3":D$=A$+B$+C$
                        na afloop is D$="BASICODE-3"
```

Zie voor stringoperatoren tevens bij LEFT$, MID$ en RIGHT$

4.7.3 De logische operatoren

Onder logische operatoren verstaan we díe operatoren die als resultaat van de operatie een logische waarde, namelijk 'waar' of 'onwaar' afgeven. Deze logische waarde kan meteen gebruikt worden, bijvoorbeeld na IF, maar kan ook worden bewaard in een numerieke variabele. Let echter op: een numerieke variabele waarin een logische waarde is opgeslagen, mag niet worden gebruikt voor rekenkundige bewerkingen.

= Vergelijkt de twee variabelen of uitdrukkingen links en rechts van het = teken met elkaar. Het resultaat is ofwel 'waar', ofwel 'onwaar'.

(Tevens: kent de waarde van de uitdrukking rechts van het = teken toe aan de variabele links van het = teken.) Voorbeelden:

```
A=(5=6)              na afloop is    A='onwaar'
IF A$=B$ THEN ...
```

< Vergelijkt twee variabelen of uitdrukkingen met elkaar en kijkt of de linker kleiner is dan de rechter. De uitkomst is een logische variabele. Indien strings vergeleken worden, wordt gekeken of de linker string eerder dan de rechter in een alfanumeriek gesorteerde rij voorkomt. (Zo kunt u dus alfabetisch sorteren.) Voorbeelden:

```
A=5:B=(A<7)                   B is nu 'waar'
A$="HO":B$="HA":A=(A$<B$)     A is nu 'onwaar'
```

\> Idem als bij <, alleen wordt nu getest op groter dan, resp. later in de rij.

<>Kijkt of twee variabelen of uitdrukkingen ongelijk aan elkaar zijn.
De uitkomst is weer een logische waarde. Voorbeelden:

```
A=(6<>7)                          A is nu 'waar'
A$="HO":B$="H":A=(A$<>B$)         A is nu 'waar'
IF A<>5 THEN ...
```

<= Kleiner dan of gelijk aan. Voor de werking zie <, maar lees in plaats van 'kleiner dan': 'kleiner dan of gelijk aan'.

\>= Groter dan of gelijk aan.

Zie voor logische operatoren verder bij `AND`, `OR` en `NOT`.

Bij de laatste drie operatoren is uitsluitend de hierboven aangegeven volgorde van de tekens toegestaan. Dus bijvoorbeeld `IF A =< 5` is FOUT.

4.8 De opbouw van een BASICODE-3 programma

Deze paragraaf geeft de voorgeschreven indeling voor de regelnummers van een goed BASICODE-3 programma:

0 - 999: De standaardroutines (zie hoofdstuk 5). Deze routines zijn per computer anders en worden door het vertaalprogramma geleverd.

1000: De eerste regel van het BASICODE-3 programma. Deze is verplicht van de volgende vorm:

```
1ØØØ A=<waarde>:GOTO 2Ø:REM programmanaam
```

<waarde> is het maximaal aantal karakters dat door alle strings te zamen gebruikt wordt. Door de sprong naar regel 20 reserveren de computers die dat nodig hebben geheugenruimte voor de strings.

1010 - 19999: Het hoofdprogramma

20000 - 24999: Subroutines die u beslist nodig hebt voor het programma, maar waarin opdrachten staan die in BASICODE-3 niet zijn toegestaan en evenmin via de subroutines realiseerbaar zijn. Deze opdrachten moeten in REM-regels worden voorzien van de nodige uitleg, zodat anderen ze aan hun computer kunnen aanpassen.

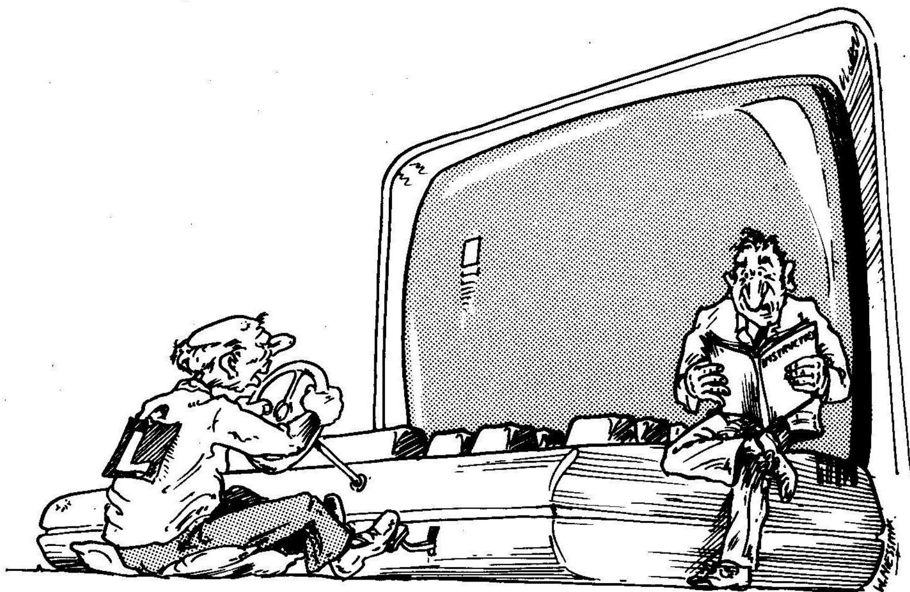
25000 - 29999: Eventueel benodigde DATA-regels.

30000 - 31999: Uitsluitend REM-regels. Hierin kan een korte beschrijving van het programma staan (indien nodig) en eventuele literatuurverwijzingen.

32000 - 32767: In REM-regels de naam en het adres van de maker van het programma.

Over de subroutines van 20000-24999 merken we nog het volgende op: probeer zoveel mogelijk te vermijden dat u opdrachten gebruikt die niet zijn toegestaan in BASICODE-3. In sommige gevallen is dat echter niet te voorkomen (bijvoorbeeld als u per se op een of andere manier met kleur wilt werken).
In dat geval zet u deze routines op de regels 20000-24999. Het is dan van groot belang dat u met REM-opdrachten goed aangeeft wat er precies in die routines moet gebeuren!

Het is verder een goede gewoonte om de regelnummers met stappen van 10 op te laten lopen, zodat later nog regels tussengevoegd kunnen worden.



5 Gebruiksaanwijzingen per merk

ACORN BBC Modellen B, B+64K en B+128K

Het vertaalprogramma wordt ingelezen als een BASIC programma. De eigenlijke vertaalroutines bestaan echter uit machinetaal. Daarom bevatten de eerste regels in een aantal REM-instructies de machinecode voor het eigenlijke vertaalprogramma. Deze regels kunnen niet gelist worden; als u het probeert hangt de machine zich op.

Het vertaalprogramma is geschikt voor de BBC-computer Model B en beide Modellen B+. Het programma is niet geschikt voor gebruik op een BBC-MASTER. Een versie voor dit type computer is in voorbereiding.

Het vertaalprogramma kan met de opdracht `LOAD ""` of `CHAIN ""` vanaf cassette worden geladen als een normaal BASIC-programma. (Bij diskmachines is het dan nodig, tevoren `*TAPE` in te tikken). Indien het met `LOAD` is ingelezen kan het meteen na het inlezen worden weggeschreven naar een eigen diskette of cassette met de gebruikelijke `SAVE` opdracht, zo nodig voorafgegaan door `*DISK`; geef daarna de opdracht `RUN`. Na enkele ogenblikken komt de prompt '>' terug en staan boven in het scherm de instructies. Met ESCAPE wordt omgeschakeld naar normaal BASIC en vandaar kan met de rode functietoets `fØ` het vertaalprogramma weer worden opgeroepen waarbij op het scherm dezelfde instructies terugkomen.

Het opnemen van de machinecode in de regels van het BASIC-programma heeft tot gevolg dat het normaal LISTen van het complete BASIC-programma niet mogelijk is. Als u per ongeluk toch een LIST-opdracht geeft, en daardoor de computer hebt laten vastlopen, kunt u de computer met `BREAK` en `OLD` weer tot de orde roepen. U kunt dergelijke problemen voorkomen door niet zomaar een LIST-opdracht te geven, maar pas beginnen te LIST-en met regel 10 of regel 1000, dus door de opdracht `LIST 1Ø,` of `LIST 1ØØØ,` (let op de komma!) te gebruiken.

Regel 0 is wel leesbaar (met `LIST Ø`) en bevat o.a. een CALL opdracht die de nodige instelling voor BASICODE verzorgt.

Oproepen van het lees-/schrijfprogramma

De f0-toets, ingedrukt vanuit BASIC, start het BASICODE-3 vertaalprogramma en geeft dan de volgende mogelijkheden (waarvan f0 en f1 op het scherm worden vermeld):
De keuzemogelijkheden

f0 : vanaf cassette inlezen van een BASICODE-3 programma, waarbij automatisch de subroutines worden klaargezet; het programma meldt het lezen met 'Searching' resp. 'Loading' zoals de BBC normaliter ook doet en tijdens het zoeken en laden ziet u de inhoud van de cassette op het scherm.

f1 : wegschrijven van een BASIC-programma als BASICODE-3, dus zonder de subroutines met regelnummers kleiner dan 1000. De schrijfroutine vraagt u allereerst, op te geven welk programma moet worden weggeschreven. U kunt kiezen uit wegschrijven van het programma in het werkgeheugen (RAM) of een programma op het achtergrondgeheugen (File). Dat laatste kan dus diskette, cassette en zelfs Econet zijn, afhankelijk van de mogelijkheden van uw computer en het door u tevoren met `*TAPE`, `*DISK`, `*ADFS` of een soortgelijk commando ingestelde 'Filing System'. Kiest u met een druk op de toets 'R' voor 'RAM' dan begint het wegschrijven meteen; kiest u met een druk op de toets 'F' voor 'File', dan vraagt het vertaalprogramma de naam van het weg te schrijven programma; na het laden daarvan begint het wegschrijven naar BASICODE-cassette. Voorafgaand aan het eigenlijke wegschrijven ziet u het hele programma snel over het scherm lopen. Het vertaalprogramma voert daarbij enkele controles uit. Meteen daarna start het schrijven van de cassette.

f2 : inlezen van BASICODE, zonder klaarzetten van subroutines; dit is nuttig wanneer een programma uit een aantal stukken bestaat: bij het eerste stuk worden de subroutines geladen, bij de volgende stukken niet;

f3 : voor het direct inlezen van de cassette, zonder op de code STX (hex 02) te wachten. Dit is nuttig bij verminkte bandjes of bij opname van een slechte radio-zender;

f4 : voor het wegschrijven van een compleet BASIC-programma in het aloude maar soms nog voor speciale toepassingen gebruikte BASICODE-1 formaat, dus inclusief de opdrachten in de regels 10 t/m 999 en zonder controle op niet-toegestane BASIC-opdrachten. Druk alleen op deze toets als u er zeker van bent dat alles wat in het geheugen staat normaal BASIC is. Als u op f4 drukt met het vertaalprogramma in het geheugen, dan zal de machine zich ophangen.

Bij dit alles is het van belang te bedenken dat de machinecode in de regels 0 t/m 9 is opgenomen : als één van die regels wordt overschreven door intoetsen of inlezen van de cassette zal het vertaalprogramma in het algemeen stranden! Ook het gebruik van `RENUMBER` moet achterwege blijven!

Bestanden in BASICODE-3

Tijdens het gebruik van BASICODE-3 ziet u op de onderste regel van het scherm aanwijzingen zoals 'zet de cassetterecorder op OPNAME/WEERGAVE' en tijdens het lezen of schrijven van een blok loopt de tekst mee op deze regel. U kunt daaraan zien dat informatie van/naar de cassette wordt getransporteerd.

Als uw cassetterecorder geen afstandsbediening voor de motor heeft, dan moet u er zelf voor zorgen dat de cassette loopt zolang de cassette-LED op de computer aan is.

Alleen bij het schrijven naar de cassette wordt telkens gewacht op RETURN voordat er iets wordt verstuurd.

Bij het inlezen van een bestand wordt gezocht naar het eerste blok (dat nummer 00 heeft). Als een blok met een ander nummer wordt gevonden, ziet u de melding '`?Blok=xx`', waarin xx het (hexadecimale) bloknummer is, dat gelezen werd. Spoel in zo'n geval de band een stukje terug en speel opnieuw af.

Voor gevorderden

Een aantal functies van het vertaalprogramma is direct toegankelijk gemaakt via een BASIC-opdracht in de vorm `CALL PAGE+xx` :

`CALL PAGE+41` : de relocator, BBC-controle, programmeren van f0:
- zorgt voor het aanpassen van de machinecode aan de feitelijk waarde van `PAGE` ingeval dat nog niet eerder gebeurde (de "relocator"),
- controleert of de computer een BBC (en dus niet een ELECTRON) is,
- stelt functietoets-0 in op `CALL PAGE+8Ø`, het startadres van de code.

Deze aanroep gebeurt automatisch na `CHAIN` of `RUN`.

`CALL PAGE+8Ø` : start vertaalprogramma (als f0 vanuit BASIC)
`CALL PAGE+83` : inlezen van BASICODE-3 (als f0)
`CALL PAGE+86` : wegschrijven van BASICODE-3 (als f1)
`CALL PAGE+89` : open een BASICODE-bestand (A%=0 lezen, anders schrijven)
`CALL PAGE+92` : lees (A%=0) of schrijf (A%<>0) een blok van een BASICODE-bestand
`CALL PAGE+95` : sluit het lezen (A%=0) of schrijven van het bestand af
`CALL PAGE+98` : zet alle 'resident integers' @%...Z% op 0

Bijzonderheden van BASICODE-3 in de BBC-computer

De maat van het tekstscherm is 25 regels van 40 karakters (MODE 7).

Het grafische scherm omvat 256 lijnen van 320 beeldpunten (MODE 4).

In het grafische scherm zijn 32 regels van 40 karakters mogelijk.

Het geluid omspant ruim vijf oktaven, namelijk van `SP=46` (ruim 1 oktaaf onder de centrale C) tot `SP=11Ø` (ruim 4 oktaven boven de centrale C).

De functietoetsen worden voor de besturing van het vertaalprogramma gebruikt en mogen bij het werken met een BASICODE-3 programma niet voor andere doeleinden gebruikt worden. Verder werken alle toetsen normaal.

Het vertaalprogramma kan worden afgebroken met BREAK. Dit is niet destructief, zodat men hierna opnieuw de betreffende functie kan proberen. Via OLD kan dan het aanwezige programma weer worden opgeroepen.

Een BASICODE-3 programma kan worden afgebroken via ESCAPE als in het programma de escape-mogelijkheid aanstaat of via BREAK als de escape-mogelijkheid niet aanstaat. Via OLD kan dan het aanwezige programma weer worden opgeroepen.

Het vertaalprogramma kan verwijderd worden door CTRL+BREAK te geven. Een eventueel aanwezig BASICODE-3 programma zal hiermee dan ook verloren gaan.

Voor machine-specifieke vragen kan de gebruiker (uitsluitend schriftelijk, gefrankeerde retourenvelop bijsluiten) terecht bij:
Big Ben Club, Vereniging van Acorn-gebruikers
Postbus 585
5201 AN 's-Hertogenbosch

ACORN ELECTRON

Inlezen van het vertaalprogramma

Het vertaalprogramma is een BASIC-programma en wordt geladen met
`CHAIN"<naam>"` en start dan automatisch of met
`LOAD"<naam>"` en daarna `RUN`.

Tijdens het inlezen, verschijnen de gebruikelijke meldingen met bloknummers (zie daarvoor de Electron User Guide).

De werking van het vertaalprogramma

Na starten toont het programma een handleiding. De handleiding kan met de SHIFT-toets doorgeschoven worden. Na afloop van de handleiding wordt het programma herstart met een druk op BREAK voor een cassettemachine of SHIFT+BREAK voor een disk-machine. Deze herstart zorgt voor het afschermen van een deel van het geheugen en het beschermen van het vertaalprogramma. Tevens wordt de handleiding gewist en worden de `FUNC`-toetsen geprogrammeerd. Na de gebruikelijke melding na een BREAK zal nu ook een beknopt overzicht van de besturing van het vertaalprogramma verschijnen.

Wordt het programma in een andere computer geladen dan een Acorn Electron, bijvoorbeeld in een Acorn BBC, dan zal bij het opstarten de melding
`??Acorn Electron??` verschijnen.

In het programma heeft men drie elementaire functies die bestuurd worden via de `FUNC` toetsen 1, 2 en 3 (weergegeven in het overzicht na BREAK):

`FUNC1` laadt een BASICODE-3 programma in het geheugen vanaf een tape
`FUNC2` schrijft een BASICODE-3 programma uit het geheugen naar een tape
`FUNC3` zet de voor BASICODE-3 benodigde hulproutines klaar

Deze elementaire functies kunnen als volgt gebruikt worden.

Inlezen van een BASICODE-3 programma

Zet met FUNC3 de hulproutines klaar.

Zet de aanlooptoon van het in te lezen BASICODE-programma voor. Druk dan op FUNC1. Tijdens het inlezen wordt het programma op het scherm getoond.

Is het programma foutloos gelezen dan wordt het vervolgens omgezet in Electron BASIC. Voor elke omgezette regel wordt een punt afgedrukt.

Zijn er fouten opgetreden dan verschijnt de melding 'Error(s)'. Het gedeelte dat ingelezen is, kan met LIST bekeken worden. Is er niet genoeg ruimte in het geheugen dan verschijnt de mededeling 'Line space' en wordt het lezen afgebroken. Treden er tijdens het lezen interpretatiefouten op dan zal tussen de BASICODE-regels de foutmelding 'Mistake' verschijnen. Het laden gaat echter gewoon door.

Wegschrijven van een programma in BASICODE

Zet een lege cassette klaar en laad het weg te schrijven BASIC programma.

Druk op CTRL+0 om de eventueel ingesteld staande bladzijdenmodus uit te schakelen.

Druk op FUNC2. Het programma wordt nu gecontroleerd op protocolfouten.
Waar nodig wordt het programma aangepast aan het BASICODE-3 protocol (bijvoorbeeld de spatie na het regelnummer). Voor elke gecontroleerde regel wordt een punt afgedrukt.

Voldoet het BASICODE-programma aan de BASICODE-3 standaard dan verschijnt de tekst 'Record then RETURN'.

Drukt men nu op RETURN dan zal het programma in BASICODE-3 formaat op cassette worden weggeschreven. Toetst men een ander teken in dan zal het vertaalprogramma stoppen. Deze optie kan benut worden om tijdens het ontwikkelen van een programma bepaalde delen te testen op protocolfouten.

Tijdens de protocolcontrole kunnen de volgende foutmeldingen optreden:

KEYWORD betekent dat op de aangegeven regel een BASIC-functie niet is toegestaan (zie de protocolhandleiding voor de toegestane BASIC-functies).

ASCII betekent dat op de aangegeven regel een ASCII-teken is gebruikt op een plaats waar dit niet mocht (bijvoorbeeld een variabelenaam met een kleine letter).

LENGTH betekent dat de aangegeven regel te lang is.

1ØØØ? betekent dat er geen regel met nummer 1000 in het BASIC-programma aanwezig is. Deze regel is verplicht in elk BASICODE-3 programma.

De BASICODE-3 hulproutines

Deze subroutines zijn nodig bij elk BASICODE-3 programma. Ze worden klaargezet door op FUNC3 te drukken. Een eventueel aanwezig programma wordt dan gewist.

Kopiëren van het programma

Een kopie van het vertaalprogramma kan gemaakt worden door het vertaalprogramma met LOAD"<naam>" te laden en vervolgens met SAVE"<naam>" weg te schrijven.

Speciale bijzonderheden bij de Acorn Electron

De maat van het tekstscherm is 25 regels van 40 karakters (MODE 6).

Het grafische scherm omvat 256 lijnen van 320 beeldpunten (MODE 4).

In het grafische scherm zijn 32 regels van 40 karakters mogelijk.

Het geluid omspant ruim vijf octaven, namelijk van SP=46 (ruim 1 octaaf onder de centrale C) tot SP=110 (ruim 4 octaven boven de centrale C).

Het £ teken zal worden weergegeven voor ASCII-code nummer 96.

De FUNC-toetsen worden voor de besturing van het vertaalprogramma gebruikt en mogen bij het werken met een BASICODE-3 programma niet voor andere doeleinden gebruikt worden. Verder werken alle toetsen normaal.

Het vertaalprogramma kan worden afgebroken met BREAK of (SHIFT+)BREAK. Dit is niet destructief, zodat men hierna opnieuw de desbetreffende functie kan proberen. Via OLD kan dan het aanwezige programma weer worden opgeroepen.

Een BASICODE-3 programma kan worden afgebroken via ESCAPE als in het programma de escape-mogelijkheid aanstaat of via (SHIFT+)BREAK als de escape-mogelijkheid niet aanstaat. Via OLD kan dan het aanwezige programma weer worden opgeroepen.

Het vertaalprogramma kan verwijderd worden door CTRL+BREAK te geven. Een eventueel aanwezig BASICODE-3 programma zal hiermee dan ook verloren gaan.

Voor machinespecifieke vragen kan de gebruiker (uitsluitend schriftelijk, gefrankeerde retourenvelop bijsluiten) terecht bij:

H.Mulder
J.W. Frisostr. 129
2713 CC Zoetermeer

APPLE II

Dit BASICODE-3 vertaalprogramma is geschikt voor de volgende computers:

- Apple 2 met Applesoft in ROM of 'Languagekaart'
- Apple 2+
- Apple 2e
- De meeste Apple-compatibles

Het vertaalprogramma maakt gebruik van de standaard-cassetterecorderaansluiting op de Apple. Voor zover aanwezig dient de printerinterface in slot 1 te zitten. Hoewel niet strikt noodzakelijk voor de goede werking, is het DOS 3.3 Operating-System en ten minste 48 K RAM gewenst.

Bij het werken met bestanden op cassette in BASICODE-formaat is het handig een hulpschakelingetje te bouwen dat de motor van de cassetterecorder aan- en uitschakelt. Indien deze schakeling is aangesloten, wordt hiervan automatisch door het vertaalprogramma gebruik gemaakt. Het schema van de schakeling vindt u aan het einde van dit hoofdstuk.

Het laden van het vertaalprogramma

Zoek op de A-kant van de cassette het vertaalprogramma op voor de Apple (luister naar de aankondiging) en zet de band op scherp. Laad het programma vanuit Applesoft zoals gebruikelijk met het commando `LOAD`.

Als het inladen gelukt is, is het verstandig vóór het runnen het vertaalprogramma met het `SAVE`-commando te saven op disk of op een andere cassette.

Na het tikken van `RUN` wordt het vertaalprogramma in het geheugen opgeborgen en is klaar voor gebruik. Het programma blijft daar staan totdat de commando's `FP` of `INT` getikt worden of het systeem opnieuw opgestart wordt met `PR#6`.

Werken met het vertaalprogramma

Het vertaalprogramma wordt aangeroepen door het tikken van & RETURN. U komt dan in het hoofdmenu waaruit u door middel van het tikken van een letter een keuze kunt maken. U hebt de volgende mogelijkheden:

`L` - lees BASICODE-3 programma in van band
`S` - schrijf BASICODE-3 programma op band
`E` - einde, keer terug naar Basic

De eerste twee opties worden nader onder de loep genomen.

Lees BASICODE-3 programma in van band

Bij de keuze inlezen, wordt het Basic-programma in het geheugen van de computer gewist, waarna de BASICODE-3 subroutines klaar worden gezet en de melding `'Start de recorder en druk op een toets'` volgt. Druk pas op een toets bij het begin van de piep aan het begin van het programma. Tijdens het inlezen van het programma knippert een karakter rechtsonder op het scherm, zodat u kunt zien dat het programma werkt. Eventueel kan het inlezen worden afgebroken door het tikken van CTRL-C.

Als het inlezen foutloos is verlopen, loopt de listing van het programma over het scherm en kan het programma gewoon door middel van `RUN` gestart worden of eerst worden gesaved op disk. Het programma kan dan in het vervolg direct van disk geladen worden, zij het dat het BASICODE-vertaalprogramma in het geheugen moet staan om het te kunnen runnen.

Indien er tijdens het inlezen een fout is opgetreden, komt er een foutmelding op het scherm en een nieuw menu:

`L` - list tabel
`O` - zet tabel om naar Basic
`E` - einde, ga naar hoofdmenu

De eerste optie list de tabel op het scherm om te kijken of er veel leesfouten zijn opgetreden. De listing is te starten en te stoppen met de spatiebalk, het indrukken van de ESC-toets stopt de listmode.

Als het aantal fouten in de listing mee blijkt te vallen, kan met de tweede optie de tabel toch worden omgezet naar Basic, waarna eventuele fouten gecorrigeerd kunnen worden. Als het aantal fouten groot is, kan beter met de derde optie worden teruggegaan naar het hoofdmenu om het inlezen nogmaals te proberen met een andere instelling van de recorder.

Schrijf BASICODE-3 programma op band

Hiermee kunt u zelf een BASICODE-3 programma opnemen op cassette (vanaf regel 1000). Voordat het feitelijke schrijven op band begint, wordt het programma gecontroleerd op een aantal belangrijke zaken, zoals de regellengte. Als er fouten worden gevonden, worden de regelnummers van de programmaregels die de fouten bevatten op het scherm weergegeven waarna deze gecorrigeerd dienen te worden.

Indien geen fouten gevonden worden, verschijnt de melding
`'Start de recorder en druk op een toets'`
op het scherm. Zet nu de recorder op opnemen en druk op een toets zodra de opname moet beginnen.

Uitgebreid hoofdmenu

Door in het hoofdmenu als keuze op de ESC-toets te drukken, verschijnt een uitgebreid hoofdmenu met als extra mogelijkheden:

M - merge BASICODE-3 programma
R - BASICODE-3 subroutines klaarzetten
W - schrijf gehele programma op band

De eerste mogelijkheid leest een programma in BASICODE van band en voegt de ingelezen programmaregels toe aan het programma dat reeds in het geheugen aanwezig is.

De tweede optie wist het programma in het geheugen en zet de BASICODE-3 subroutines klaar, zodat u zelf een programma in BASICODE-3 kunt schrijven.

Met de derde optie wordt het gehele programma in het geheugen in BASICODE op de band geschreven, dus inclusief alle regels onder de 1000. Dit is in tegenstrijd met het BASICODE-protocol, maar het is in sommige speciale gevallen nuttig.

De BASICODE-3 mogelijkheden van de Apple

- afmeting tekstscherm: 24 regels van 40 tekens (BASICODE-norm)
- grafische resolutie: 256 punten horizontaal, 192 punten verticaal
- tekst op grafisch scherm: 24 regels van 42 tekens
- geluid: SV=Ø geluid uit
  SV>Ø geluid aan
- Tonen hoger dan SP=115 worden weergegeven met SP=115. Het geluid wordt tegelijkertijd weergegeven via de luidspreker en de aansluiting van de cassetterecorder.

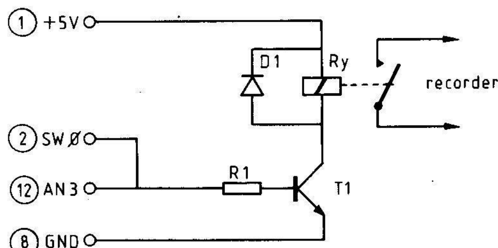
- de subroutine voor het uitschakelen van de STOP-toets doet niets.

De hulpschakeling voor de cassetterecorder

Onderstaande schakeling maakt het werken met bestanden eenvoudiger, omdat niet telkens met de hand de recorder gestart en gestopt hoeft te worden.

Met het maakcontact van het relais wordt de motor van de cassetterecorder aan- en uitgeschakeld (vooropgesteld dat uw recorder deze mogelijkheid heeft).

De schakeling wordt aangesloten op de Game-I/O aansluiting in de Apple via een 16-pens IC-voetje. De vermelde nummers slaan op de pinnummers van de Game-I/O voet.



Onderdelenlijst

R1	4K7 Ohm
D1	IN4148
T1	BC548
Ry	reedrelais 5 V, 1 x maakcontact

Voor vragen over en speciale problemen met betrekking tot het vertaalprogramma kan men zich wenden tot de auteur van het programma (uitsluitend schriftelijk, met bijsluiting van een gefrankeerde retourenvelop):
J. Herrmann
p/a Postbus 1410
5602 BK Eindhoven

COMMODORE 64

Inlezen van het vertaalprogramma

Het vertaalprogramma is geheel in machinetaal geschreven en het kan dus niet met een LIST-opdracht worden uitgelezen. Het wordt na het resetten (uit- en aanzetten) van de computer ingelezen vanaf cassette met de opdracht LOAD of desgewenst met LOAD"C-64 BASICODE-3". Tijdens het inlezen is er helaas niets op het scherm te zien. Het inlezen van de cassette duurt bijna drie minuten. Na het inlezen kan het programma worden gestart met RUN.

Na deze RUN-opdracht brengt het programma de computer in de BASICODE-toestand. Daarbij wordt de geheugenruimte van $8C00 t/m $CFFF in gebruik genomen en voor BASIC afgeschermd. Dit duurt slechts één seconde. Vervolgens verschijnt de melding dat de machine in de BASICODE-toestand is gebracht en staan in het geheugen alleen de vaste BASICODE-3 subroutineregels.

Het is erg waarschijnlijk dat het vertaalprogramma niet zal werken in een C-64 met andere toebehoren dan een originele Commodore-diskdrive en/of printer. Met name zal het programma niet werken indien een cartridge is ingestoken. Wordt het programma in een andere Commodore-computer geladen dan een C-64 dan zal het eveneens niet werken maar vrijwel zeker de hele computer op 'tilt' brengen, en zal waarschijnlijk alleen een reset die computer weer tot een normaal gedrag brengen.

Het programma werkt niet op een Commodore 128 in '64-mode'. De auteur stelt het op prijs als een eigenaar en kenner van de Commodore 128 de oorzaak daarvan en een geschikte oplossing ervoor weet te vinden.

Gebruik van het vertaalprogramma

Nadat de computer in de BASICODE-toestand is gekomen, kan men een keuzemenu oproepen door op één van de functietoetsen (maakt niet uit op welke) te drukken. In dat menu staat dan vermeld dat men met een druk op functietoets f3 een BASICODE-programma kan inlezen (waarbij indien nodig de vaste subroutines automatisch worden klaargezet), en dat men met een druk op functietoets f5 een reeds aanwezig programma, dat aan het BASICODE-3 protocol voldoet, kan wegschrijven.

Met een druk op de spatiebalk wordt het menu weer verlaten en keert men terug naar het normaal functioneren van de computer.

Als men, terwijl het menu op het scherm staat, op f2 drukt, verschijnen enkele extra regels in beeld waaruit blijkt dat vervolgens met functietoets f2 de standaard-subroutines worden klaargezet (nuttig als men zelf een programma in BASICODE wil schrijven), dat met functietoets f4 BASICODE wordt ingelezen en achter het eventueel reeds aanwezige programma wordt aangekoppeld (let er dan op dat dan wel de regelnummers verder oplopen), en dat met functietoets f6 alle programmaregels (dus vanaf regelnummer 0) in BASICODE-formaat worden weggeschreven. De laatste twee mogelijkheden zijn alleen van belang indien men volgens de thans geheel verouderde BASICODE-1 standaard wenst te werken.

Kiest men voor inlezen van BASICODE dan dient de recorder te worden gestart. Daartoe verschijnt de gebruikelijke melding op het scherm. Daarna wordt het gehele scherm rood. Zodra het leesprogramma een BASICODE-aanlooptoon leest, wordt het scherm groen. Zodra het eigenlijke programma wordt ingelezen, zal het beeldscherm elke 1,2 seconde van kleur veranderen totdat het hele programma ingelezen is. Daarna verschijnt de normale READY-melding. Indien een leesfout is geconstateerd wordt dat na het inlezen gemeld. Het programma kan dan toch gelist worden, waardoor men mogelijk de leesfout kan vinden en herstellen. Als kleurveranderingen uitblijven, wordt blijkbaar niet correct ingelezen. Men kan het inlezen dan afbreken door de recorder te stoppen, de band terug te spoelen, en via enkele drukken op functietoetsen het inlezen opnieuw te proberen.

Als men in het menu kiest voor functietoets f5 (of f6), dus voor wegschrijven, dan zal het vertaalprogramma eerst controleren of de weg te schrijven programmaregels aan de belangrijkste eisen voldoen, namelijk of de regellengte maximaal 60 tekens is en of er in de programmaregels verboden commando's voorkomen (zoals bijv. POKE en SYS). Deze controle gaat erg snel en is op het scherm zichtbaar. Als een fout wordt gevonden, stopt het programma met een melding in welke regel welke fout is geconstateerd. Men dient dan zelf die regel te listen en te verbeteren. Daarna kán opnieuw met de juiste functietoets het wegschrijven worden opgeroepen.

Als geen fouten worden gevonden, komt meteen na de controle de normale melding dat men de recorder op opnemen dient te zetten. Daarna start het wegschrijven. Tijdens het op cassette wegschrijven van een BASICODE-programma zal net als tijdens het inlezen het beeldscherm na elke 1,2 seconde van kleur wisselen. Alleen tijdens het schrijven van de 5 seconden durende aanlooptoon is het scherm constant groen gekleurd.

Het vertaalprogramma kopiëren

Een kopie van het vertaalprogramma kan gemaakt worden door het vertaalprogramma eerst met LOAD in te lezen, vooral nog geen RUN-opdracht te geven maar juist een passende SAVE-opdracht, bijvoorbeeld
SAVE"C-64 BASICODE-3" of
SAVE"Ø:C-64 BASICODE-3",8 als men over een diskdrive beschikt.
Kopiëren van het programma is uitsluitend toegestaan voor eigen gebruik.
Zie in dit verband de Appendix over copyright.

Speciale bijzonderheden bij de Commodore 64:

In BASICODE bestaat het tekstscherm uit 25 regels van elk 40 karakters. Grafische tekens zijn niet toegestaan; onder BASICODE werkt de machine trouwens uitsluitend in de tekststand, waarbij hoofdletters naast kleine letters mogelijk zijn.

Het grafische scherm omvat 200 lijnen van 288 beeldpunten. Het grafische scherm is dus iets smaller dan wat maximaal mogelijk zou zijn.

In het grafische scherm zijn 25 regels tekst van maximaal 36 karakters per regel mogelijk.

Het geluid omspant theoretisch bijna negen octaven, namelijk van `SP=Ø` tot en met `SP=1Ø6`. De laagste twee octaven produceren echter veeleer bromtonen dan wat men als aangenaam geluid ervaart.

De in de meeste computers beschikbare accolades zijn in de C-64 niet mogelijk. In plaats daarvan komen blokhaken op het scherm.

De functietoetsen worden uitsluitend voor het oproepen van het BASICODE-menu gebruikt. De kleurtoetsen mogen in BASICODE niet gebruikt worden.

Het vertaalprogramma kan tijdens het lezen of schrijven van cassette worden afgebroken door de recorder te stoppen. In noodgevallen kan men STOP+RESTORE gebruiken.

Een BASICODE-3 programma kan worden afgebroken met de STOP-toets. Het verdient dan aanbeveling meteen het commando `GOTO95Ø` in te tikken om zodoende de computer in de normale toestand te schakelen. Als het programma de STOP-toets heeft uitgeschakeld (subroutine 280), zal in de meeste gevallen ook STOP+RESTORE niet werken, en is afbreken alleen mogelijk door het uitschakelen van de computer.

In de C-64 is het lezen en schrijven van bestanden naar cassette of diskette (anders dan in BASICODE-formaat, dus met `NF=2` of groter) niet geheel volgens het protocol uitgevoerd. In alle gevallen wordt namelijk `IN=Ø` afgeleverd. De auteur van het vertaalprogramma houdt zich aanbevolen voor suggesties tot verbetering van de desbetreffende subroutines.

De BASICODE-3 subroutines bevatten veel `SYS`-opdrachten naar programma's in machinetaal in het afgeschermde geheugengebied. Een BASICODE-programma zal dus alleen kunnen werken als in dat geheugengebied inderdaad de desbetreffende programmatuur aanwezig is. Breng dus altijd eerst de computer in de BASICODE-toestand alvorens een BASICODE-programma te gebruiken, dus lees eerst het vertaalprogramma in en `RUN` dit alvorens een BASICODE-3 programma te laden en te starten. Dit is vooral van belang indien men ingelezen BASICODE-programma's op eigen diskette opslaat en ze later weer vanaf schijf inleest.

Voor vragen over en speciale problemen met betrekking tot het vertaalprogramma kan

men zich wenden tot de auteur van het programma (uitsluitend schriftelijk, met bijsluiting van een gefrankeerde retourenvelop):
J. Haubrich
p/a Postbus 1410
5602 BK Eindhoven

EXIDY SORCERER

Het vertaalprogramma is 'all in', dus het bevat de BASICODE-lees- en -schrijfroutines alsmede alle benodigde subroutines, waarvan sommige in Basic en andere in machinetaal zijn geschreven.

Inlezen:
a. Basic ROM Pac moet in de computer aanwezig zijn en de cassetterecorder moet op de Exidy-manier zijn aangesloten.
b. Laden vanaf cassette gaat NIET met `CLOAD` maar via de Monitor, dus: typ in `BYE` RETURN
c. Typ in `LOG` , zet de recorder op afspelen en druk dan pas op de returntoets.

Zodra het programma (`BC3SB`) gevonden is, wordt dit op het scherm gemeld met de mededeling dat nu geladen wordt. Zodra het laden klaar is, wordt ook dat gemeld en verschijnt het woord 'READY'. Typ `RUN 9ØØ` RETURN om te starten.

Na dit startcommando verschijnt er een menu. Zie daarover hieronder.

Het programma is geschikt voor een EXIDY SORCERER van elke geheugengrootte, doch om er goed mee te kunnen werken, wordt tenminste 32K en nog liever 48K geheugengrootte aanbevolen, want het vertaalprogramma zelf is ruim 9K groot.

De werking van het programma

Het programma werkt alleen als een BASICODE-interface volgens onderstaand schema voorhanden is. Het programma werkt ook als het eerder genoemd ROM Pac voorzien is van de Basic EXTension (BEXT), hoewel een speciaal daarop toegespitste versie (BC3BX) bij de aan het einde genoemde instantie verkrijgbaar is.

De keuzemogelijkheden via het menu zijn:
De hoofdfuncties:
1 Een BASICODE-programma van cassette inlezen
2 Het programma in BASICODE naar cassette schrijven
3 Programma beëindigen

De speciale funkties:
4 Het eigenlijke gebruiksprogramma wissen, dus de programmaregels 1000 en hoger wissen
5 Alle Basicregels wissen om een 'klassiek' BASICODE-1 programma te kunnen maken (in BASICODE wegschrijven naar tape is dan slechts mogelijk door in te

typen: `POKE 26Ø,126:POKE 261,1Ø:A=USR(A):LIST` RETURN)
6 Inlezen van een BASICODE-1 programma, dus een programma dat geen subroutines gebruikt
7 Het gebruiksprogramma in Exidy-formaat wegschrijven vanaf regel 1000 (tapesparend want zonder 9K routines!)
8 In Exidy-formaat inlezen van een met punt 7 weggeschreven gebruiksprogramma achter de reeds gereedgezette routines

Na intoetsen van het gewenste nummer zal de computer uw keuze uitvoeren. Het in de derde alinea genoemde interface moet voor de acties 1, 2, 5 en 6 tussen de parallel-uitgang van de computer en de recorder zijn aangebracht.

Een kopie maken

Een kopie van het vertaalprogramma maken kan op twee manieren. Verwijder indien nog een gebruiksprogramma aanwezig is eerst de regels 1000 en volgende via het menu (`RUN 9ØØ`).

a Wegschrijven op tape: De Exidy-kabels aan- of omsteken op de standaard Exidy cassette-interface. Typ in
`BYE` RETURN
`SE X=ØC78` RETURN
`SE F=42` RETURN
`DU Ø1B7 Ø1B8` RETURN (op 01B7 staat xx
op 01B8 staat yy)
`SA BC3SB Ø1B7 yyxx`

Let op de volgorde!
Zet de recorder nu op opname en druk pas dán de RETURN-toets.

b Voor bewaren op diskette typt men eerst
`PRINT PEEK(44Ø)` RETURN
waarna het aantal (nn) pages opgegeven wordt. Vervolgens
`BYE` RETURN
`GO Ø` (of koude start zoals u gewend bent uw diskdrive op te starten)
RETURN
`SAVE nn BC3SBPAC.COM` RETURN

Bijzonderheden van BASICODE-3 in de Exidy

Het tekstscherm omvat 30 regels met 64 tekens per regel.

Het grafische scherm bestaat uit 90 lijnen met 112 punten per lijn.

In het grafische scherm zijn maximaal 29 tekstregels van 55 tekens mogelijk, maar dan is het gehele grafische scherm overschreven.

De laagst mogelijke toon komt overeen met SP=50 (de D onder de centrale C), de hoogste toon komt overeen met SP=93 (de derde A boven de centrale C).

Het 'at'-teken (@, het zogenaamde aapje) vernietigt de regel waarin dit teken gebruikt wordt. Het inleesprogramma wijzigt het veiligheidshalve in #.

BASICODE-3 'verstaat' het gebruik van de pijltjestoetsen gelijk ingedrukt met de shifttoets, zoals u dat gewend bent.

Met CTRL-C is het vertaalprogramma en ook een gebruiksprogramma af te breken.

Hoewel het protocol spaties voorschrijft voor sommige opdrachten, zoals GOTO, hoeft men zich hierover weinig zorg te maken - het vertaalprogramma regelt dit. Wel moet u er op bedacht zijn dat door het vertaalprogramma eventueel ingebrachte spaties meetellen voor de maximale regellengte van 60 tekens.

Verder gelden de volgende afwijkingen:

Geaccentueerd printen (routine 150) is slechts zichtbaar door een grafisch bolletje voor en na het desbetreffende woord.

Grafische tekst is alleen mogelijk in voorgrondkleur (CN=Ø) waarbij het onderliggende grafische teken verdwijnt.

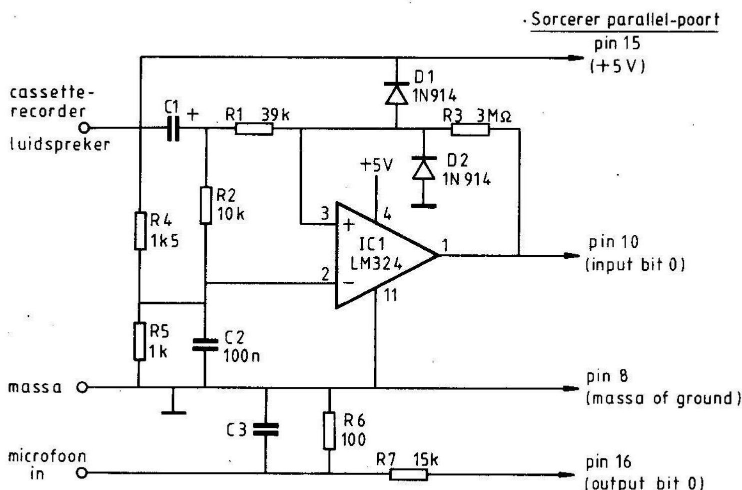
Let voorts op het al dan niet gebruik van het interface tussen recorder en Sorcerer. Bij het in de 4de alinea besproken menu werd hierop al uw aandacht gevestigd. Ook bij BASICODE-bestandsverwerking moet u dit interface gebruiken. Wegens aanwezigheid van genoemd interface is het dan overigens niet mogelijk gelijktijdig bij BASICODE-bestandsverwerking de printer te gebruiken, tenzij u zelf wachtlussen in het programma aanbrengt om pluggen te wisselen.

De programmatuur veronderstelt cassette-motorcontrol, maar er zijn voldoende waarschuwingen ingebouwd om de recorder met de hand te kunnen bedienen.

Wanneer de gebruiker speciale vragen mocht hebben, kan hij informeren wie hem kan helpen bij de

EXIDY SORCERER GEBRUIKERS GROEP
Secretariaat Prins Hendrikstraat 3d
3071 LG Rotterdam telef.010-4330493
aan welk adres ook de speciale BEXT-versie en CP/M-versies BASICODE-3 voor EX- en MBASIC te bestellen zijn.

schema 1



Voor de BASICODE-lees- en -schrijfroutines is een stukje hardware nodig omdat we niet de standaard Exidy cassette-interface gebruiken. We doen dit met bit 0 van de parallelinterface. Voor schrijven gebruiken we een verzwakker om de cassetterecorder niet te oversturen.

Dit niveau is ongeveer 35 mVtt hetgeen overeenkomt met de microfooningang van de meeste cassetterecorders. Zoniet gebruik dan de 'line'-ingang, die is minder gevoelig.

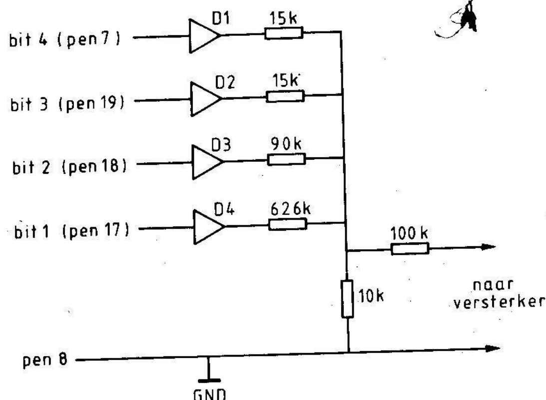
Voor het inlezen gebruiken we een Schmitt-trigger om van de analoge signalen een ttl-signaal te maken. De hier gebruikte interface heeft bewezen zeer betrouwbaar te zijn (ook bij andere computers).

Voor BASICODE-3 was het nodig de hardware wat uit te breiden (zie schema 2). Op bit 1 t/m 3 van de parallel-outputpoort is een digitaal/analoog-convertor gemaakt om volume te kunnen realiseren voor de muziekroutines. Echter deze d/a-convertor is niet absoluut noodzakelijk. Dezelfde toon is ook beschikbaar op bit 0 van de outputpoort. Echter hier is de volumeregeling vervallen. Je kunt hier de interface van schema 1 voor gebruiken.

Bij beide interfaces (schema 1 en 2) is voor de muziek een versterker nodig. Het meest eenvoudige is hiervoor de cassetterecorder te gebruiken en dan zonder de automatische volumeregeling, anders zijn we alsnog de volumeregeling kwijt. Je kunt natuurlijk ook een aparte versterker maken.

Het eenvoudigste is de d/a-convertor in een aparte plug onder te brengen. De weerstandswaarden zijn 1% nauwkeurig omdat ik die toevallig had liggen. Als u de dichtstbijliggende waarden van de 5% reeks gebruikt, zult u geen enkel verschil horen maar wel goedkoper uit zijn.

schema 2
parallel outputpoort



D1 t/m D4 kan elke willekeurige (klein signaal) diode zijn, bijv.:BAW62, 1N4001.

MSX-COMPUTERS SPECTRAVIDEO SV.318 EN SV.328

Er zijn twee verschillende vertaalprogramma's, één voor de Spectravideo SV.328 en SV.318 met 64K RAM, de andere voor alle MSX-machines met tenminste 64K RAM, zowel MSX-1 als MSX-2. De gebruiksaanwijzing is voor alle gevallen dezelfde en wel als volgt.

Inlezen van het vertaalprogramma

Het vertaalprogramma kan met de volgende procedure worden ingelezen:

- computer uit en weer aan;
- tik: `BLOAD "CAS:"` (en ENTER);
- na 'Found: BASICO' en 'Ok' even controleren of het goed ingelezen is met het volgende BASIC-regeltje (vergeet ENTER niet):

```
T=Ø: FOR A%=&H88ØØ TO &HBFFF: T=T+PEEK(A%): NEXT: PRINT T
```

Na ongeveer één minuut moet het getal 1287903 op het scherm verschijnen. Ieder ander getal betekent dat het vertaalprogramma ondanks de 'Ok' toch niet correct ingelezen is.

Kopiëren van het programma

Het vertaalprogramma kan met één van de volgende opdrachten (naar disk of een andere cassette) weggeschreven worden:

```
BSAVE "CAS:Basico", &H88ØØ, &HCØØØ (+ENTER) (cassette)
BSAVE "1:Basico", &H88ØØ, &HCØØØ   (+ENTER) (SV.328 disk)
BSAVE "Basico", &H88ØØ, &HCØØØ     (+ENTER) (MSX disk)
```

Na inschakelen van de computer kan het vertaalprogramma met één van de volgende opdrachten opnieuw geladen en gestart worden:

```
BLOAD "CAS:Basico",R (+ENTER)    (cassette)
BLOAD "1:Basico",R   (+ENTER)    (SV.328 disk)
BLOAD "Basico",R     (+ENTER)    (MSX disk)
```

Bij het inlezen van cassette dient de computer "Found: Basico" op het scherm af te drukken en vervolgens verschijnt na enige tijd de introductiepagina, die na ca. 15 seconden (of na het indrukken van een toets) verdwijnt.

De werking van het programma

Wanneer het vertaalprogramma niets anders te doen heeft, wordt het hoofdschemamenu weergegeven. De gebruiker kan met één de toetsen A t/m Q een programmadeel selecteren en via de ESC-toets activeren; het is altijd mogelijk het geactiveerde programmadeel met CTRL + STOP te onderbreken. Door op ENTER te drukken, wordt het extra menu actief, waarmee het tekstgeheugen en het BASIC-geheugen gewist en alle REM-regels uit het BASICODE-programma verwijderd kunnen worden. Het geselecteerde programmadeel wordt boven in 't beeld schematisch weergegeven. De selecties zijn:

A dit programmadeel leest een BASICODE-programma van cassette, vertaalt dat programma en schakelt het BASIC-ROM in ('Ok'). Het zojuist vertaalde BASICODE-programma kan met (C)SAVE weggeschreven of met RUN gestart worden;

B via dit programmadeel wordt een BASIC-programma naar BASICODE vertaald en op cassette weggeschreven. Het te vertalen programma moet wel eerst ingelezen worden, wat op twee manieren gebeuren kan.

Wanneer het te vertalen programma met een SAVE"...",A (ASCII-SAVE) weggeschreven is, kan het door het vertaalprogramma via programmakeuze M (cassette) of O (disk) ingelezen worden. In het andere geval moet men het vertaalprogramma via Q verlaten, het BASIC-programma met (C)LOAD laden en het vertaalprogramma met CMD en ENTER weer opstarten. Nu kan het BASIC-programma via schema B vertaald en naar cassette geschreven worden;

C dit programmadeel leest een BASICODE-programma van cassette zonder

het te vertalen. Het is dan voor verdere bewerking in het 'tekstgeheugen' beschikbaar;

D het BASICODE-programma in het tekstgeheugen wordt naar cassette geschreven;

E deze programmadelen verzorgen het inlezen/wegschrijven van

F BASICODE-3 bestanden van/naar het tekstgeheugen, zodat ook de bestanden eenvoudig bewerkt kunnen worden;

G het BASICODE-programma in het tekstgeheugen wordt vertaald en het vertaalprogramma wordt verlaten (als schema A, echter zonder inlezen);

H een BASIC-programma wordt door de BASICODE-vertaler vertaald en naar het tekstgeheugen gebracht (schema B zonder wegschrijven);

I de inhoud van het tekstgeheugen wordt zonder vertaling met het momentele BASIC-programma gemengd, waarna het vertaalprogramma verlaten wordt (eventueel eerst BASIC-geheugen wissen);

J een BASIC-programma wordt zonder vertaald te zijn aan het tekstgeheugen toegevoegd (eventueel eerst tekstgeheugen wissen);

K dit is de ingebouwde editor waarmee het tekstgeheugen bewerkt kan worden. Het editorscherm bevat 22 regels van 37 tekens per regel; boven in beeld is het aantal vrije en gebruikte bytes te zien. De cursor is vrij over het scherm te bewegen, waarbij het indrukken van een toets leidt tot het veranderen, invoegen of verwijderen van één of meer tekens in de tekst. Het einde van een regel (ENTER) wordt door een speciaal grafisch teken weergegeven. Wanneer een naar BASICODE vertaald BASIC-programma bekeken wordt, zullen eventuele protocolfouten geïnverteerd weergegeven worden en door de cursor op een fout te zetten, geeft de functietoetsregel aan welke fout gevonden is. Na het indrukken van de ESC-toets wordt de editor verlaten;

L hiermee kan de inhoud van het tekstgeheugen op een printer afgedrukt worden. Eerst dienen enkele printerinstellingen opgegeven te worden (of steeds op ENTER drukken). Door de BASICODE-3 vertaler gevonden protocolfouten worden onderstreept;

M met deze programmadelen kan de inhoud van het tekstgeheugen van of

N naar cassette gelezen of geschreven worden, als bij de normale LOAD- en SAVE-opdrachten;

O de equivalenten van de LOAD- en SAVE-opdrachten van of naar disk;

P waarbij eerst een inhoudsopgave gegeven wordt;

Q hiermee kan het vertaalprogramma verlaten (naar BASIC overgeschakeld) worden. Met CMD + ENTER wordt het vertaalprogramma weer gestart.

De uitvoering van BASICODE-3 op deze computers:

Het tekstscherm bestaat uit 24 regels van 40 tekens.

Het grafische scherm omvat 192 lijnen van elk 192 beeldpunten.

Tekst op grafisch scherm: 24 regels van 32 tekens (SV.328)
24 regels van 24 tekens (MSX).

Het geluid beslaat 8 octaven, namelijk de waarden van SP=21 tot en met SP=116. Te lage of te hoge tonen worden 1 octaaf (12 toonnummers) verschoven.

Wanneer de onderbreekfunctie niet verboden is, kan het BASICODE-programma met CTRL+STOP onderbroken worden.

Voor vragen en een uitgebreide handleiding (f 5,-):
Spectravideo/MSX C.U.C.,
Postbus 202,
2300 AE Leiden.

P2000M - CP/M-VERSIE

Onder CP/M kan een BASICODE-programma van cassette in de P2000M worden ingelezen en in een programma worden omgezet. Daarna kan het met Disk-BASIC worden gestart en gebruikt. Tevens kan een BASIC-programma volgens het BASICODE-systeem op cassette worden geschreven. Daartoe dient u op schijf te hebben staan:

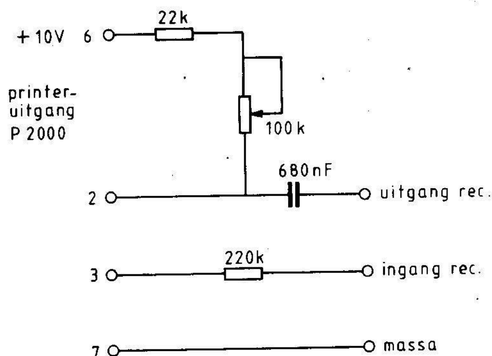
```
BASICODE.COM
BC3     .BAS
FUNCTION.DAT
KEYWORDS.DAT
BC3     .COM
```

alsmede uiteraard een goede Disk-BASIC zoals:
MBASIC.COM of BASIC8Ø.COM.

Een schijf met deze programma's, subroutines en data-files kunt u krijgen bij één van de gebruikersgroepen; in dat geval zijn enkele van de volgende alinea's niet van belang. Hebt u een cassetteband, dan moet u de data-files daarvanaf op schijf zetten en van de .HEX-files programma's maken met behulp van LOAD.

De Cassette-interface

Bouw een interface volgens het schema op de volgende bladzijde:



Intypen TESTAFR.HEX

Type met behulp van ED of WORDSTAR de volgende routine in:

```
:1ØØ1ØØØØ119FØ1CD96Ø1F3ØE8Ø3ECØD31ØØ64CCD59
:1ØØ11ØØØ9CØ13E4ØD31ØDBØØEEFFC246Ø1Ø649CDF4
:1ØØ12ØØØ9CØ1ØDC2Ø9Ø13ECØD31ØØ624CD9CØ13EA6
:1ØØ13ØØØ4ØD31ØDBØØEEFFC246Ø1Ø621CD9CØ1ØD2D
:1ØØ14ØØØC226Ø1C3Ø7Ø111ØBØ2CD96Ø1F316F3215C
:1ØØ15ØØØØØØØØ6ØØ5C3E2Ø1278E63FC6ØØ5F673E66
:1ØØ16ØØØ2A12Ø6ØØØ42824DB2ØE6Ø128F7ØØØØØØFC
:1ØØ17ØØØ5D3E2Ø1278E63FC65Ø5F6F3E2A12Ø6ØØB1
:1ØØ18ØØØØ428Ø8DB2ØE6Ø12ØF718C9DBØØEEFFCACF
:1ØØ19ØØØ54Ø1FBC3ØØØØØE Ø9CDØ5ØØC91ØFEC9ØCB7
:1ØØ1AØØØ7ØDØA546573742Ø656E2Ø616672656779
:1ØØ1BØØØ656C2Ø726F7574696E652Ø76616E2Ø636Ø
:1ØØ1CØØØ6173736574746520696E746572666163CA
:1ØØ1DØØØ65ØDØAØA4E65656D2Ø6E752Ø65656E2Ø99
:1ØØ1EØØØ7Ø726F65667374756B6A652Ø6F7Ø2EØD23
:1ØØ1FØØØØA41666272656B656E2Ø6D65742Ø656C8Ø
:1ØØ2ØØØØ6B652Ø746F6574732E2Ø24ØCØ7537Ø6522
:1ØØ21ØØØ656C2Ø6865742Ø7Ø726F65667374756BA9
:1ØØ22ØØØ6A652Ø61662EØDØA526567656C2Ø6E75E1
:1ØØ23ØØØ2Ø64652Ø696E7465726661636520706FØ5
:1ØØ24ØØØ746D2E2Ø656E2Ø6465ØDØA766F6C756D79
:1ØØ25ØØØ652Ø6B6E6F7Ø2Ø76616E2Ø64652Ø63612F
:1ØØ26ØØØ73736574746520726563F726465720D73
:1ØØ27ØØØØA7A6F2Ø61662Ø6461742Ø64652Ø73745B
:1ØØ28ØØØ657272656E2Ø28696E2Ø6865742Ø6265EB
:1ØØ29ØØØ656C6429ØDØA7A6F2Ø676F65642Ø6D6F45
:1ØØ2AØØØ67656C696A6B2Ø6F6E6465722Ø656C6B44
:1ØØ2BØØØ61617220737461616E2E2EØDØAØA4166627B
:1ØØ2CØØØ72656B656E2Ø6D65742Ø656C6B652Ø745E
```

```
:1ØØ2DØØØ6F6574732Ø6B616EØDØA616C732Ø6572BB
:1ØØ2EØØØ2Ø6765656E2Ø696E7Ø75742Ø69732E2ØB5
:1ØØ2FØØØ24F36F2Ø61662Ø6461742Ø64652Ø737448
:ØØØØØØØØØØ
```

N.B.: Bovenstaande getallen zijn hexadecimaal, dus Ø=NUL.
Save dit op schijf.

Maak er een .COM-file van door in te typen `LOAD TESTAFR` RETURN.
Indien een typefout is gemaakt, geeft `LOAD` een foutmelding. Herstel de fout en `LOAD` opnieuw.

Afregelen

Start `TESTAFR.COM` door in te typen `TESTAFR` RETURN

Het afregelen gaat dan als volgt:

Het programma geeft eerst een in hoogte variërende toon af. Zet deze toon enkele minuten lang op de cassetteband. De toon kan met elke toets worden afgebroken.

Speel dan de toon terug van de band. U ziet bij het tweede deel van het programma bovenin het beeld vier sterren.

Zorg er voor dat de sterretjes zo dicht mogelijk onder elkaar komen. Afbreken kan door de recorder te stoppen en een willekeurige toets in te typen.

Intypen file BC3READ.HEX

Type met behulp van ED of WORDSTAR de volgende routine in:

```
:1ØØ1ØØØØ11ØAØ2CDØ3Ø2ØEØ1CDØ5ØØD92EØØD9F34C
:1ØØ11ØØØ3E4ØD31ØDBØØEEFF2ØFA119EØ2CDØ3Ø219
:1ØØ12ØØØ1E2621AAØ2AF7716ØØ7AØ8Ø6ØØ5ØCDC518
:1ØØ13ØØØØ1DACØØ1Ø882BB38FØØ6Ø8CDDFØ1CB1917
:1ØØ14ØØØ1ØF979FE823AØ9Ø22ØØ1AFA932Ø9Ø27939
:1ØØ15ØØØEE8Ø462377D926F2772CD978FEØ32ØC784
:1ØØ16ØØØ3AØ9Ø2E6FF28Ø61182Ø2CDØ3Ø21141Ø27C
:1ØØ17ØØØCDØ3Ø2ØE16115CØØCDØ5ØØ21AAØ27EFEØ1
:1ØØ18ØØØØ2232ØFA23118ØØØØ68Ø7E231213FEØD25
:1ØØ19ØØØ28Ø218Ø32B36ØAFEØ328Ø91ØEDE5CDFAD4
:1ØØ1AØØØØ1E118E11B3E1A12131ØFACDFAØ1ØE1ØEC
:1ØØ1BØØØ115CØØCDØ5ØØ1154Ø2CDØ3Ø2FBC3ØØØØØ9
:1ØØ1CØØØ116EØ218F4C5DB2ØØ6ØØ4F14DB2ØB92ØA5
:1ØØ1DØØØØB1ØF8DBØØEEFF28F237C1C9AFC1C9C56B
:1ØØ1EØØØØ6Ø216ØØCDC5Ø138ØDCDC5Ø138Ø87ABB11
:1ØØ1FØØØ3ØØ21ØEEC1C9C1C118C6ØE15115CØØCD88
:1ØØ2ØØØØØ5ØØC9ØEØ9CDØ5ØØC9ØØØC424333524513
```

```
:1ØØ21ØØØ41442Ø2876657273696520312E32290D9C
```

N.B. Bovenstaande getallen zijn hexadecimaal, dus 0=nul.

Save dit op een schijf.

Maak er een .COM-file van door in te typen LOAD BC3READ en geef RETURN. Indien u een typefout hebt gemaakt, geeft LOAD een foutmelding.
Herstel de fout en load opnieuw.

Start BC3READ.COM door in te typen BC3READ BASICODE.HEX RETURN. Op het scherm komt te staan 'Start recorder en geef RETURN'. Zoek op de band de eerste van de vijf files op voor de P2000M. Zet de recorder op weergave en druk de toets RETURN in. Op het scherm is na een paar seconden te zien, dat er wat ingelezen wordt. Geeft BC3READ een foutmelding, controleer dan de interface en de afregeling en probeer het nog een keer.

BC3READ zal de file op schijf zetten onder de naam BASICODE.HEX
BC3READ is nu niet meer nodig en kan van de schijf verwijderd worden.

Maak van BASICODE.HEX een .COM-file, (zie hierboven)

Met BASICODE.COM worden de volgende vier files ingelezen:

Start BASICODE.COM door in te typen BASICODE en geef RETURN. Na enkele seconden verschijnt er een menu op het scherm.

Kies Nr.2. Nu staat er op het scherm 'Wat is de naam van de in te lezen file'. Antwoord met in te typen FUNCTION.DAT RETURN. Dit wordt ingelezen en op schijf gezet. Als een leesfout gemeld wordt, is dit op te lossen door opnieuw in te lezen of met ED te kijken wat de fout is en zo mogelijk verbeteren. Als er geen leesfout is geconstateerd, drukken we op een willekeurige toets, waarna het menu verschijnt.

Doe hetzelfde als in de vorige alinea, maar dan met de volgende file op de band en geef het de naam KEYWORDS.DAT

Doe nogmaals hetzelfde maar weer met de volgende file op de band en geef het de naam BC3.BAS

Doe ten derden male hetzelfde met weer de volgende file op de band en geef het de naam BC3.HEX. Ga nu terug naar CP/M en maak van BC3.HEX een .COM-file, (zie hierboven).

BASICODE.COM is nu gereed voor gebruik.

BASICODE is een menu-geörienteerd programma, waarmee men BASICODE-files kan inlezen van banden en op schijf zetten, spaties en BASICODE-subroutines kan toevoegen, lezen van schijf en in BASICODE op band zetten. Start dit programma door in te typen BASICODE RETURN. Er verschijnt een menu op het scherm.

Deze programmatuur is geschikt voor P2000M met CP/M. Voor andere CP/M-machines zal het niet direct te gebruiken zijn, omdat er machine-afhankelijke eigenschappen zijn gebruikt, maar het is misschien wel aan te passen voor andere CP/M-machines.

Het keuze-menu

Inlezen

Tik nu cijfer 2 in, waarna de tekst verschijnt die vraagt een filenaam in te tikken. Voeg aan deze naam .BAS toe en geef RETURN. Start nu de recorder en wacht de beginfluittoon af. Zodra deze te horen is, geeft u weer een RETURN. Op het scherm worden drie regels zichtbaar waarin u de ingelezen data kunt zien. Het inlezen is te onderbreken door de recorder te stoppen en een toets in te drukken. Na het inlezen volgt de mededeling:

```
File ingelezen

<FILENAAM>BAS is op disk gezet

U kunt het thans van spaties voorzien
```

Spaties toevoegen

Kies nu cijfer 5 als u de file van subroutines wilt voorzien, anders cijfer 4. Geef RETURN. Nu verschijnen de inhoudsopgave en een mededeling die vraagt welke file u van spaties wilt voorzien. Vul <FILENAAM> in met de toevoeging BAS. Geef een RETURN.

```
Na de tekst: KEYWORDSDAT geladen
             FUNCTIONDAT geladen
             BC3         BAS geladen (alleen bij 5)
             PROGNAAMBAS geladen
```

ziet u de data van onder naar boven scrollen. Als de file van spaties is voorzien, wordt het op schijf gezet.
De boodschap :
<PROGNAAM> is op disk gezet
Druk een toets in om verder gaan

verschijnt. Doe dit laatste om terug te keren in menu.

Schrijven naar cassette

Maak de cassetterecorder gereed voor opname. Kies in het menu cijfer 3. Kijk in de inhoudsopgave en geef de file-naam op en geef RETURN. Na regellengte-controle krijgt u de opdracht om de cassetterecorder aan te zetten en geeft u weer: RETURN
N.B.: Het bovenstaande veronderstelt dat u met ASCII ge-SAVE-de .BAS files werkt.

Starten

Opstarten van een ingelezen BASICODE-programma:
In het menu Nr.0 kiezen (terug naar CP/M)
Daarna ofwel: intypen BC3
of: intypen BC3 programmanaam

BC3 zet een stuk machinetaal in het geheugen en start MBASIC op.
Als men na BC3 geen programmanaam had opgegeven, dient men vervolgens in te tikken:
LOAD "PROGNAAM.BAS" en als de subroutines er niet voor gezet zijn
LOAD "BC3.BAS" en MERGE "PROGNAAM.BAS".

Bijzonderheden van BASICODE-3 in de P2000M

Het tekstscherm beslaat 24 regels van 80 tekens.

Het grafische scherm omvat 72 lijnen van 152 beeldpunten per lijn.

In het grafische scherm is plaats voor 24 regels van 76 tekens.

SD kan variëren van 2 tot en met 100.
SP kan variëren van 21 tot en met 100.

De karakters 92, 95, 96, 123, 125, 126 en 127 wijken af van normaal.

Afbreken is mogelijk met ^C

Subroutine 650 overschrijft grafische karakters.

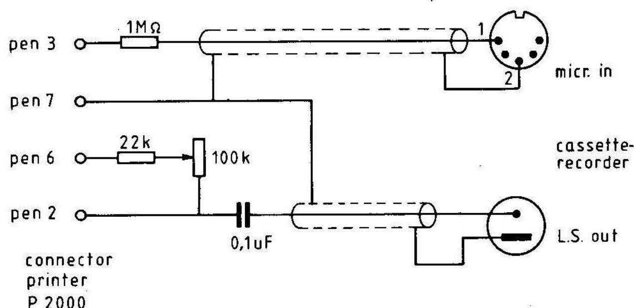
Met dank aan: Henk Warnitz, Rob de Beer, Hermine Bakker en Toon Kooyman
Henk Koorn

PHILIPS P2000T

Voor het in de computer inlezen van het P2000 BASICODE-vertaalprogramma zijn twee dingen noodzakelijk, namelijk:

1. Een audio-cassetterecorder en interface
2. Het schrijf-/leesprogramma

De P2000 heeft standaard geen aansluiting voor een cassetterecorder. Via de printeraansluiting kan echter toch een cassetterecorder worden aangesloten. De hiervoor benodigde interface/aansluitkabel is bij de P2000 gebruikersvereniging te koop, maar wie dar niet tegenop ziet moet even de soldeerbout ter hand nemen om dit hulpschakelingetje te maken:



Benodige onderdelen:
1. 25-pins RS232 kabel-connector (mannetje)
2. 5-polige DIN kabelstekker 180 graden (mannetje)
3. 2-polige DIN luidsprekerstekker (mannetje)
4. Instelpotentiometer 100 kΩ - $^1/_2$ W
5. Weerstand 1 MΩ - $^1/_4$ W
6. Weerstand 22 kΩ - $^1/_4$ W
7. Polyestercondensator 0,1 µF
8. 2-aderig afgeschermd snoer van circa 1 $^1/_2$ m

Deze onderdelen zijn verkrijgbaar bij elke goede elektronicahandel. Men kan de onderdelen onderbrengen in de kap van de stekker die op de printeraansluiting komt. Allereerst moet de potentiometer zó afgeregeld worden dat de kans op fouten bij transmissie zo klein mogelijk wordt. Met deze instelweerstand wordt de ingangsgelijkspanning van de P2000T bepaald.
Dat kan met het volgende hulpprogramma.

Instellen cassette-interface

Het afstellen van de potentiometer in de interface is eenvoudig uit te voeren met het programma in de nu volgende listing:

```
 1Ø REM Afregeling cassette interface
 2Ø CLEAR 1ØØ,&H9FØØ
1ØØ REM Proef-tonen opnemen
11Ø DATA 9F2Ø
12Ø DATA F3,ØE,8Ø,3E,CØ,D3,1Ø,Ø6,4C,CD
13Ø DATA 62,9F,3E,4Ø,D3,1Ø,DB,ØØ,EE,FF
14Ø DATA C2,6Ø,9F,Ø6,49,CD,62,9F,ØD,C2
15Ø DATA 23,9F,3E,CØ,D3,1Ø,Ø6,24,CD,62
16Ø DATA 9F,3E,4Ø,D3,1Ø,DB,ØØ,EE,FF,C2
17Ø DATA 6Ø,9F,Ø6,21,CD,62,9F,ØD,C2,4Ø
18Ø DATA 9F,C3,21,9F,FB,C9,1Ø,FE,C9,XX
19Ø RESTORE 11Ø : S=-16194 : GOSUB 5ØØ
2ØØ DEFUSR1=A : PRINT CHR$(12)
21Ø PRINT"Neem nu een proefstukje op."
22Ø PRINT"Afbreken met elke toets."
23Ø A=USR1(Ø) : PRINT CHR$(12)
24Ø PRINT "Nu weerstand afregelen"
3ØØ REM Sterren boven in het beeld
31Ø DATA 9F8Ø
32Ø DATA F3,16,5Ø,26,ØØ,2E,ØØ,Ø6,ØØ,5C
33Ø DATA 3E,2Ø,12,78,E6,3F,C6,ØØ,5F,67
34Ø DATA 3E,2A,12,Ø6,ØØ,Ø4,28,24,DB,2Ø
35Ø DATA E6,Ø1,28,F7,ØØ,ØØ,ØØ,5D,3E,2Ø
36Ø DATA 12,78,E6,3F,C6,5Ø,5F,6F,3E,2A
37Ø DATA 12,Ø6,ØØ,Ø4,28,Ø8,DB,2Ø,E6,Ø1
38Ø DATA 2Ø,F7,18,C9,FB,C9,XX,
39Ø RESTORE 31Ø : S=-19559 : GOSUB 5ØØ
4ØØ DEFUSR2=A
41Ø PRINT CHR$(12): A=USR2(Ø): GOTO41Ø
42Ø END
5ØØ REM Zet machinetaal in geheugen
51Ø READ P$ : I=VAL("&H"+P$) : A=I : T=I
52Ø READ P$ : IF P$="XX" THEN 55Ø
53Ø P=VAL("&H"+P$) : POKE I,P : I=I+1
54Ø T=T+P : GOTO 52Ø
55Ø IF T=S THEN RETURN
56Ø PRINT "TYPEFOUT GEMAAKT" : END
```

U kunt het programma letterlijk overtypen (let op: in de DATA-regels is een Ø het cijfer 0, niet de letter O; het zijn immmers hexadecimale getallen). Als u het programma laat lopen wordt een eenvoudige controle op typefouten uitgevoerd. De afstelling gaat als volgt:

1. Het programma geeft eerst een in hoogte variërende toon af. Zet deze toon enkele minuten lang op de cassetteband. De toon kan met elke toets worden afgebroken.

2 Speel dan de toon terug van de band. Zet het volume van de cassetterecorder maar goed hard. U ziet bij het tweede deel van het programma bovenin het beeld sterren, ongeveer als volgt:

```
*    *
*    *
```

Draai nu het volume van de cassetterecorder met kleine stapjes terug. Als de sterren niet meer bewegen, draai dan aan de regelweerstand tot dit wel weer zo is. Herhaal deze twee handelingen tot de stand van de regelweerstand is gevonden waar bij minimaal volume de sterren nog nét bewegen. Draai dan het volume omhoog tot de sterren mooi stabiel boven elkaar staan. Afbreken met de STOP-toets. Later kan het afregelen gemakkelijk nog eens herhaald worden met BASICODE-vertaalprogramma waar een soortgelijke testroutine in opgenomen is.

Het schrijf-/leesprogramma

Als de cassette-interface goed is afgeregeld, kunt u het nu volgende schrijf-/leesprogramma gaan intikken. Dit gaat op dezelfde manier als bij het afregelprogramma.

```
1000 CLEAR 50,&H9DFF : PRINT CHR$(12)
1010 G$=CHR$(130) : W$=CHR$(135)
1020 PRINT CHR$(2);G$;"Even geduld";W$;
1030 READ P$ : I=VAL("&H"+P$) : T=0
1040 READ P$ : IF P$="XX" THEN 1080
1050 P=VAL("&H"+P$) : POKE I,P : I=I+1
1060 IF I MOD 12 = 0 THEN PRINT ".";
1070 T=T+P : GOTO 1040
1080 PRINT CHR$(12) : IF T+I=10632 THEN 1120
1090 PRINT CHR$(129);CHR$(157);CHR$(141);W$;
1100 PRINT SPC(7);"Typefout gemaakt !"
1110 PRINT : END
1120 DEF USR8=&H9E08 : DEF USR9=&H9E28
1130 PRINT G$;"Met";W$;"?usr8(0)";G$;
1140 PRINT "zet u een P2000-programma"
1150 PRINT SPC(13);G$;"op de audio-cassette"
1160 PRINT : PRINT G$;"Met";W$;"?usr9(0)";G$;
1170 PRINT "leest u een programma van"
1180 PRINT SPC(13);G$;"audio-cassette";
1190 PRINT G$;"in de P2000" : END
2000 DATA 9E00
2010 DATA 00,00,00,00,00,00,00,00,B7,2A,05,64
2020 DATA ED,5B,5C,62,ED,52,22,04,9E,CD,5C,9E
2030 DATA 36,55,3E,40,CD,8E,9E,CD,69,9E,3E,20
2040 DATA CD,8E,9E,C9,D5,CD,5C,9E,CD,E6,9E,20
2050 DATA 17,2A,5C,62,ED,5B,04,9E,19,22,05,64
2060 DATA 22,07,64,22,09,64,CD,69,9E,CD,E6,9E
2070 DATA E1,36,00,C8,00,36,06,23,36,56,23,36
```

```
2080 DATA 9E,C9,81,45,72,72,6F,72,21,04,9E,22
2090 DATA 00,9E,21,06,9E,22,02,9E,C9,2A,5C,62
2100 DATA 22,00,9E,2A,05,64,22,02,9E,C9,3E,C0
2110 DATA 06,83,10,FE,D3,50,3D,20,F7,C9,D5,EB
2120 DATA 2A,02,9E,B7,ED,52,EB,D1,23,C9,F3,37
2130 DATA 16,00,2A,00,9E,CD,BF,9E,06,27,AE,32
2140 DATA 07,9E,7E,CD,B6,9E,CD,82,9E,06,22,3A
2150 DATA 07,9E,20,EE,06,22,CD,B6,9E,CD,76,9E
2160 DATA FB,C9,16,10,17,CD,CC,9E,20,FA,C9,06
2170 DATA 4B,CD,D1,9E,20,F9,CE,FE,38,F5,06,1F
2180 DATA CD,D1,9E,04,04,10,FE,30,04,06,30,10
2190 DATA FE,08,2F,E6,80,F6,40,D3,10,08,06,2A
2200 DATA 15,C9,F3,2A,00,9E,CD,32,9F,3E,16,CD
2210 DATA BF,9E,32,07,9E,CD,32,9F,06,20,CD,35
2220 DATA 9F,38,F9,CD,35,9F,06,35,CD,24,9F,77
2230 DATA 3A,07,9E,AE,32,07,9E,CD,82,9E,06,30
2240 DATA 20,EE,CD,24,9F,21,07,9E,BE,F5,CD,76
2250 DATA 9E,F1,FB,C9,16,08,5F,CD,32,9F,7B,17
2260 DATA 06,34,15,20,F5,C9,CD,35,9F,05,DB,20
2270 DATA 0F,A9,F2,35,9F,A9,4F,3E,80,B8,C9,XX
32000 REM P2000-programma's op audio-cassette
32010 REM is gemaakt door Frans van Dal
```

Bij RUN is er weer een controle op typefouten. Let in dat geval speciaal op de volgende dingen:

- het verschil tussen de hoofdletter O en het cijfer 0,
- het verschil tussen de hoofdletter I en het cijfer 1,
- het verschil tussen de hoofdletter B en het cijfer 8,
- een komma tussen de DATA, dus niet een punt of een spatie,
- geen extra komma aan het einde van de DATA-regels,
- let ook op de ; aan het einde van sommige PRINT-regels.

Wie al het BASICODE-1 of BASICODE-2 vertaalprogramma voor de P2000T heeft, hoeft geen onnodig typewerk te verrichten. Op de BASICODE-3 cassette is het schrijf-/leesprogramma gezet in BASICODE. U kunt het inlezen met BASICODE-1 of met BASICODE-2, het werkt in beide gevallen even goed.

Met dit schrijf-/leesprogramma kunnen we P2000 programma's in BASIC wegschrijven en inlezen met een gewone audiorecorder. Het programma reserveert een stukje geheugen waarin (in machinetaal) de schrijf- en leesroutine komt te staan.

Op de BASICODE-3 cassette staat het vertaalprogramma voor de P2000T. Het is er op gezet met het hier beschreven schrijf-/leesprogramma. Daarmee kunt u het dus ook weer inlezen in uw P2000T. Zoek het vertaalprogramma op en zet de recorder klaar op het begin van de aanlooptoon en sluit de interface aan op de luidsprekeruitgang. Ga dan als volgt te werk:

1. Type in: ?usr9(Ø) maar druk NOG NIET op ENTER.
2. Zet de recorder op weergeven.
3. Druk nu onmiddellijk de ENTER-toets in.
4. Na 10 seconden hoort u een piepje, het programma is dan gevonden.
5. Na ongeveer 1 minuut hoort u weer een piepje, het inlezen is klaar. Verschijnt daarbij de boodschap 'ERROR', dan is er iets mis gegaan. Verander het volume van de recorder en probeer nogmaals het programma in te lezen. Heeft de recorder een toonregeling, dan kan daarmee ook het inlezen worden beïnvloed.
6. Is het vertaalprogramma goed ingelezen, CSAVE het dan meteen op een mini-cassette.

Overigens kan men met dit schrijf-/leesprogramma ook zelf BASIC programma's op een audiocassette zetten. Dit gaat met andere tonen dan bij BASICODE. Voordeel is dat hiermee alle P2000 BASIC programma's kunnen worden weggeschreven, ook als er machinetaal in zit zoals in het BASICODE-3 vertaalprogramma. Wegschrijven gaat als volgt:

1. RUN eerst het schrijf-/leesprogramma.
2. CLOAD het weg te schrijven programma in de P2000.
3. Zet de cassetterecorder op opnemen.
4. Tik in: ?usr8(Ø) en druk op ENTER.
5. Na het tweede piepje staat het programma op de cassette.

Nadeel van deze routines is dat het inlezen een stuk kritischer is dan BASICODE en dat hiermee geen uitwisseling van programma's met andere merken mogelijk is.

BASICODE-menu

U kunt de BASICODE-machinetaalroutine aanroepen door middel van SHIFT+8 op het kleine toetsenbord. Er verschijnt dan het volgende menu op het scherm:

```
Menu BASICODE 3.Ø

1. Testtoon opnemen
2. Interface afregelen
3. Schrijf hele programma
4. Schrijf vanaf regel 1ØØØ
5. Lees programma
6. Lees en voeg subroutines toe
7. Inlezen na Geheugen vol
8. Voeg subroutines toe
9. Einde
   Wat kiest U?
```

We zullen enkele gebruiksmogelijkheden en procedures nader toelichten. BASICODE-3 bevat ook de afregelprocedures voor de cassette-interface.

Afregelen van de cassette-interface

1. Zet de recorder op opnemen.
2. Kies van het menu mogelijkheid 1. Er wordt een toon van 2400 Hz opgenomen.
3. Druk na enkele minuten op een willekeurige toets om te stoppen.
4. Kies Menu 2, spoel de cassette terug en speel af. Op de onderste regel verschijnen twee bewegende sterren. Draai nu de regelweerstand van de interface zó dat de sterren bij het laagste volume van de recorder nog net bewegen. Draai vervolgens de volumeregelaar op zodat de sterren op elkaar vallen.
5. Druk een willekeurige toets in om te stoppen.

Schrijven van BASICODE

1. Haal het weg te schrijven programma (met `CLOAD`) in het geheugen.
2. Zet de recorder op opnemen.
3. Kies Menu-mogelijkheid 3. U hoort een piepje en het programma wordt op de band gezet.
4. Als de boodschap '`Klaar`' op het scherm komt, stop dan de recorder. Nu staat het hele programma op de band.

Bij programma's in BASICODE-3 moeten de standaardsubroutines niet op cassette worden geschreven.

Door Menukeuze 4 te kiezen in plaats van Menukeuze 3 worden alleen de programmaregels vanaf 1000 op de band gezet.

Lezen van BASICODE

1. Zet het in te lezen programma klaar in de cassetterecorder.
2. Kies Menukeuze 5.
3. Zet de recorder op weergeven. Er verschijnen eerst de bekende sterren onderaan het beeld en dan ziet u alle ingelezen letters in hoog tempo links onder op het beeld langskomen.
4. Als het hele programma is ingelezen, LIST het zich automatisch.
5. Wanneer er geen inleesfouten zijn ontdekt, verschijnt de melding 'Klaar'. Het programma staat nu in het geheugen van de P2000.

Met punt 8 van het Menu worden de standaardroutines van BASICODE-3 toegevoegd aan het programma dat in het geheugen staat.

Door voor het inlezen punt 6 van het Menu te kiezen in plaats van 5 worden de routines meteen al voor het nog in te lezen programma gezet.

Foutmeldingen

`Leesfout:` programma is fout ingelezen. Het staat echter wel in het geheugen en kan geLIST worden. Ga na of er 'vreemde' karakters in staan, bijvoorbeeld `PR*NT`.

Hoogstwaarschijnlijk moet de '*' een 'I' zijn. U kunt het programma ook RUNnen om tegen de Syntax errors aan te lopen.

`Afgebroken`: dit gebeurt als u tijdens het inlezen of wegschrijven op een toets drukt. Kom dus niet aan het toetsenbord teneinde het schrijven en lezen niet af te breken!

Soms verschijnt de boodschap: '`Geheugen vol`'. Een programma, indien het in BASICODE op band staat, bevat meer karakters (bytes) dan hetzelfde programma wanneer dit in het geheugen staat. De opdracht `PRINT` neemt bijvoorbeeld 5 bytes in beslag in BASICODE en maar één byte in de P2000T. Hierdoor kunnen lange programma's niet in één keer ingelezen worden.

Bij bovenstaande boodschap wordt het tot nu toe ingelezen programma eerst in gecodeerde vorm opgeslagen. In de nu vrijgekomen ruimte kan de rest van het programma ingelezen worden. Hiervoor moet teruggespoeld worden naar het begin van het programma. Daarna met punt 7 van het Menu het programma opnieuw inlezen.

Men kan uit het Menu springen door middel van keuze 9.

Bijzonderheden van de P2000T in BASICODE-3

Het tekstscherm omvat 24 regels van 40 tekens

Het grafische scherm beslaat 72 lijnen van elk 78 beeldpunten

In het grafische scherm zijn 24 tekstregels van 39 tekens mogelijk

De bruikbare waarden voor `SP` variëren van 48 als minimum tot 127 als maximum. Wanneer `SP` onder 48 komt worden de tonen een of meer oktaven hoger weergegeven.

Het lezen en schrijven van BASICODE kan op elk moment worden onderbroken door op een willekeurige toets te drukken.

Een BASICODE-3 programma kan worden afgebroken met SHIFT+STOP, tenminste zolang deze niet is uitgeschakeld.

Voor de goede werking is een vrij geheugen van 31350 bytes noodzakelijk.

De mogelijkheid om op de P2000 met bestanden op disk te werken, is weggelaten, daar iedereen toch van de minstens zo gebruikersvriendelijke minicassetterecorder gebruik kan maken. Hierdoor wordt tevens geheugenruimte uitgespaard.

Frans van Dal
Yoeri Geutskens

SINCLAIR ZX-SPECTRUM

Inlezen

Lees het vertaalprogramma in met LOAD "". Het vertaalprogramma bestaat uit twee delen die na elkaar ingelezen worden: de BASIC-subroutines en het eigenlijke vertaalprogramma.

Zodra het geheel is ingelezen, start het programma vanzelf en presenteert het zich. Daarna komt u in BASIC terecht. Het programma is nu klaar voor gebruik.

Werking en keuzemogelijkheden

De diverse delen van het vertaalprogramma worden geactiveerd door het intikken van een sterretje '*', gevolgd door een letter:

*L : Lees een BASICODE-programma in.
*T : Vertaal het ingelezen programma naar Spectrum-BASIC.
*C : Converteer een programma in Spectrum-BASIC naar BASICODE.
*W : Schrijf het geconverteerde programma in BASICODE weg.
*K : LIST het onvertaalde programma.
*S : SAVE het vertaalde programma in Spectrum-formaat.
*B : Backup van het vertaalprogramma.

U mag zowel hoofdletters als kleine letters intypen. Indien u alleen een sterretje of een niet bestaand commando invoert, wordt een menu met de bovenstaande commando's getoond; u kunt dan de gewenste keuze maken door op de desbetreffende lettertoets te drukken.

*L

Hiermee leest u een BASICODE-programma in. Zet de recorder op de begintoon van de opname en stel hem in op het gebruikelijke niveau. Zodra het inleesprogramma de begintoon gevonden heeft, verschijnen er strepen op de schermrand; als het werkelijke programma ingeladen wordt, ziet u de programmaregels onderop het scherm voorbijflitsen.

Zolang het inleesprogramma geen BASICODE-signaal leest, zal de schermrand regelmatig knipperen tussen rood en cyaan.

Het inlezen stopt als u op de spatiebalk drukt of als het geheugen vol is of als het einde van het BASICODE-programma bereikt is. Indien het inlezen succesvol was verlopen, gaat het programma meteen verder met vertalen (de volgende stap), anders krijgt u een foutmelding 'BREAK into program', 'Out of memory' of 'Tape loading error'. U kunt dan eventueel alsnog vertalen door de *T-optie te kiezen.

*T

Deze optie wordt normaal na het foutloos inladen automatisch uitgevoerd. Het in het vrije geheugengebied ingelezen BASICODE-programma wordt omgezet naar een programma in Spectrum BASIC. Elke vertaalde programmaregel wordt op het scherm getoond.

*C

Dit is het omgekeerde van de optie '*T': het in de Spectrum aanwezige BASIC-programma wordt vanaf regel 1000 omgezet naar een BASICODE-programma in het vrije geheugengebied.

Allereerst kijkt het vertaalprogramma of u geen verboden opdrachten hebt gebruikt zoals bijvoorbeeld POKE, BEEP of PLOT en of alle functies gevolgd worden door een argument tussen haakjes (dus niet SIN X en TAB 1Ø maar SIN (X) en TAB (1Ø)). Mocht er toch zo'n fout in uw programma zitten dan krijgt u de melding '*ERROR*' gevolgd door de regel waarin de fout zit met een vraagteken achter het foute woord. U kunt dan deze regel verbeteren en de *C-optie opnieuw starten.

Indien het programma geen van bovenstaande fouten bevat, wordt automatisch verdergegaan met het eigenlijke vertalen. Het is hierbij mogelijk dat u een melding 'Line too long at line ' gevolgd door het regelnummer krijgt; er is dan een regel ontstaan langer dan de toegestane 60 karakters. Het omzetten zal hierdoor NIET stoppen. Indien het geheugen te klein is voor het onvertaalde en vertaalde programma tegelijk krijgt u na het omzetten de melding 'BASIC PROGRAM OVERWRITTEN'; mocht het geheugen ook te klein zijn voor het vertaalde programma alleen dan volgt een 'OUT OF MEMORY' foutmelding. In beide gevallen bent u dan het oorspronkelijke programma kwijt!

*W

Hiermee schrijft u het op bovenstaande wijze omgezette programma in BASICODE-formaat naar cassette; u kunt dit een onbeperkt aantal malen doen. Het naar cassette weggeschreven signaal wordt op de schermrand getoond zoals bij het inlezen. U kunt dit afbreken met de spatiebalk.

*K

Dit is de List-optie (Hint: het LIST-keyword zit op de K-toets). Een programma dat in BASICODE-vorm (direct na inlezen of voor wegschrijven) in het geheugen zit, kunt u normaal niet LISTen, vandaar deze optie. U kunt hiermee ook beoordelen wat er binnengekomen is na een inleesfout.

*S en *B

Met de '*S'-optie kunt u het vertaalde BASICODE-programma SAVEn in gewoon Spectrum formaat zodat u het later direct kunt inladen. Met '*B' maakt u een backup van het vertaalprogramma. U hebt bij deze opties de keuze tussen cassette en Microdrive (indien aangesloten).

Opmerking: Deze opties lopen via de BASIC-regels 960 resp. 980 zodat u ze eventueel aan uw eigen opslagsysteem kunt aanpassen.

Werken met BASICODE-3 bestanden

De bestandsroutines werken voor NF=Ø en NF=1 met BASICODE-bestanden volgens het protocol. Moet er een blok gegevens naar tape weggeschreven worden dan verschijnt op de onderste schermregel de melding 'PRESS <PLAY> & <REC> THEN ANY KEY' en wacht de Spectrum op een toetsdruk alvorens de gegevens weg te schrijven. Bij inlezen krijgt u de melding 'START TAPE..'. Indien er een blok wordt ingelezen dat voorbij het verwachte blok ligt, verschijnt de melding 'WRONG RECORD - REWIND TAPE' op het scherm.

Voor de overige waarden van NF wordt de Microdrive als opslagmedium gebruikt. Eventueel kunt u de OPEN- en CLOSE-opdrachten aan uw eigen opslagsysteem aanpassen. Ingeval van een leesfout wordt naar regel 550 gesprongen, maar dat is in het laatste geval dan niet meer gegarandeerd.

Werken met BASICODE-1

Het is met dit vertaalprogramma mogelijk, overeenkomstig de oude BASICODE-1 standaard, alle programmaregels vanaf regelnummer 2 op cassette te schrijven. Daartoe dienen twee POKE-opdrachten te worden ingetikt:

POKE 547ØØ,2 en
POKE 547Ø1,Ø.

Na deze opdrachten is het werken volgens de BASICODE-3 norm niet meer mogelijk. Men dient het vertaalprogramma dan opnieuw in te lezen.

Implementatie BASICODE-3

Het tekstscherm omvat 24 regels van 42 karakters. Een eventueel aangesloten ZX-printer of een daarmee vergelijkbare printer geeft eveneens 42 karakters per regel.

Het grafische scherm omvat 192 lijnen van elk 256 punten per lijn (dus de twee onderste schermregels doen ook mee).

Het grafische scherm kan 24 regels tekst van 42 karakters weergeven.

Het geluid omvat alle toegestane waarden van SP (0 tot 127); er is echter slechts één volume mogelijk (bij SV=Ø geeft subroutine 400 geen geluid, bij alle andere waarden wel en op dezelfde sterkte). De toon kan maximaal 10 seconden duren (SD=1Ø4).

U mag NIET het copyright-symbool (code 127) gebruiken omdat deze code in BASICODE-3 voor de DELETE-functie gebruikt wordt. De control-codes van de 42-koloms schermroutine zijn als volgt:

16 t/m 23 als normaal,
24 gevolgd door N stelt het aantal schermregels in (1...24),
25 = PRINT komma,
26 = continu scrollen,
27 = wacht voor scroll op toetsdruk,
28 t/m 31 en 127 volgens de BASICODE-3 standaard.

Bijzonderheden

Dit vertaalprogramma bevat een speciale BASIC-interpreter die de Spectrum-BASIC compatible maakt met BASICODE-3. U mag nu dus regelnummers hoger dan 9999 en string-, array- en FOR-NEXT-variabelen met namen van maximaal twee letters gebruiken. Tevens hoeft u bij string-arrays niet meer een extra dimensie op te geven en ook is een index 0 toegestaan.

De functies LEFT$, MID$ en RIGHT$ zijn aanwezig, echter in de vorm van userdefined functies, u moet ze dus als FN LEFT$ etc. intypen.
N.B.: de bij Sinclair gebruikelijke notatie A$(X TO Y) is NIET toegestaan.

ON-GOTO en ON-GOSUB zijn aanwezig in de vorm GOTO/GOSUB *V;2ØØØ,3ØØØ, 4ØØØ... in plaats van de gebruikelijke ON V GOTO/GOSUB 2ØØØ,3ØØØ,4ØØØ... (lees het sterretje als ON).

Er is nu echter wel een 'CLEAR , A' nodig (zie regel 20) waarvan de waarde afhangt van de totale geheugenruimte aan strings die u met INPUT en stringkoppeling samenstelt. Voor de meeste programma's voldoet een waarde van 100 à 250; indien u dit te klein kiest, krijgt u een 'Out of string space' foutmelding. Houd vooral met bovenstaande zaken rekening bij het schrijven van een BASICODE-3 programma; het omzetprogramma kan niet alle fouten opsporen!

Bij het gebruik van diskinterfaces anders dan de Sinclair Interface 1 kunt u moeilijkheden krijgen bij opdrachten die op het gebruik van de interface slaan; vermijd hierbij het gebruik van variabelen en typ getallen als VAL"getal" in.

N.B.: Regel 1 mag nimmer uit het vertaalprogramma verwijderd worden!
Zonder deze regel is werken met BASICODE niet betrouwbaar.

Voor eventuele problemen kunt u contact opnemen met de:
Sinclair gebruikersgroep,
Postbus 142,
1740 AC Schagen   Jan Bredenbeek

SPECTRAVIDEO SV.318 EN SV.328

Het vertaalprogramma voor de Spectravideo SV.318 en de Spectravideo SV.328 werkt op dezelfde wijze als het vertaalprogramma voor de MSX-computers. Eenvoudigheidshalve wordt daarom verwezen naar de gebruiksaanwijzing voor die computers.

Er zijn enkele minieme verschillen; deze worden hierna afzonderlijk genoemd. U vindt ze eveneens in de tekst voor de MSX-computers.

1. Na het inlezen en het starten en het controleren van het controlegetal kan het vertaalprogramma op disk of cassette worden weggeschreven met:

```
BSAVE "CAS:Basico", &H88ØØ, &HCØØØ   (+ENTER)   (cassette)
BSAVE "1:Basico", &H88ØØ, &HCØØØ     (+ENTER)   (SV.328 disk)
```

2. Na inschakelen van de computer kan het vertaalprogramma dan weer met één van de volgende opdrachten geladen en gestart worden:

```
BLOAD "CAS:Basico",R   (+ENTER)   (cassette)
BLOAD "1:Basico",R     (+ENTER)   (SV.328 disk)
```

De uitvoering van BASICODE-3 op deze computers:

Op het grafische scherm zijn 24 tekstregels van elk 32 tekens mogelijk.

Voor vragen en een uitgebreide handleiding (f 5,-) kan men terecht bij:
Spectravideo C.U.C.,
Postbus 202,
2300 AE Leiden

P. Zevenhoven

ANDERE MERKEN EN TYPEN

Zoals in het begin van dit boek vermeld, is BASICODE een ontwikkeling van een aantal actieve, geïnteresseerde hobbyisten die elk op welhaast professioneel niveau met hun computer om kunnen gaan.

Voor sommige computermerken/typen was bij het ter perse gaan van dit boek een BASICODE-3 vertaalprogramma in voorbereiding. De mogelijkheid bestaat dat dat programma bij het uitkomen van dit boek klaar is. Dit betreft de volgende computers:

IBM-PC (en soortgenoten)
Schneider CPC-464, CPC-664 en CPC-6128
Sinclair QL
Tandy TRS-80 Model 1 en Model 3

Bij de desbetreffende gebruikersclubs kunt u desgewenst informeren of het programma klaar is en hoe u het kunt verkrijgen.

BASICODE-2 bruikbaar?

Voor een aantal computers bestaat wel een BASICODE-2 vertaalprogramma maar (nog) niet een BASICODE-3 vertaalprogramma. Dankzij de uitwisselbaarheid is het zo dat elk Basicode-programma kan worden ingelezen met beide typen vertaalprogramma's. Programma's die gebruik maken van de nieuwe mogelijkheden van BASICODE-3 (zie hiervoor Appendix 2), zullen niet werken onder het BASICODE-2 vertaalprogramma. De reden is eenvoudig: de benodigde subroutines zijn niet aanwezig. In eenvoudige gevallen zullen veel hobbyisten in staat zijn zelf de benodigde subroutines aan te vullen. In moeilijker gevallen is het aan te bevelen, te informeren bij de gebruikersclub van het eigen merk.

Hoe ontstaat een vertaalprogramma?

Een vertaalprogramma voor een bepaald type computer komt pas tot stand als er uit de kring van de gebruikers van die machine iemand zich opwerpt om dat programma 'even te maken'. Dat moet dan iemand zijn die:

1 Goed op de hoogte is van de werking van zijn computer
2 In machinetaal programmeert alsof hij nooit anders heeft gedaan
3 Erg in BASICODE geïnteresseerd is
4 Van doordouwen weet
5 En tegen een beetje kritiek kan

Kortom, een schaap met vijf poten. De Stichting BASICODE is voor zo iemand het aangewezen adres. Vanuit de Stichting BASICODE kan enige steun worden geboden bij het op de juiste manier vertalen van BASICODE, bij het aanpakken van bepaalde problemen en bij het controleren of het vertaalprogramma inderdaad overeenkomt met wat BASICODE zou moeten zijn.

Als er al een BASICODE-2 vertaalprogramma voor uw type computer bestaat, dan is de stap naar BASICODE-3 meestal niet zo moeilijk meer. Helemaal vanaf 'niets' beginnen kost het nodige aan denkwerk en uitprobeerwerk. Schrik niet, want voor allerlei andere merken zijn vele enthousiastelingen u voorgegaan.

Belangstellenden voor computermerken die in dit boek niet zijn vermeld, en die zich na en ondanks bovenstaande beschrijving in staat achten zelf een BASICODE-3 vertaalprogramma te maken, kunnen bij de Stichting BASICODE informeren. Het adres is:
Stichting BASICODE Postbus 1410 5602 BK Eindhoven

6 Demonstratieprogramma's

In dit hoofdstuk geven we de listings van de BASICODE-3 programma's die op de cassette als demonstratie zijn opgenomen. Waar dat voor beter begrip van de BASICODE-3 subroutines nuttig lijkt, wordt de listing onderbroken met enkele regels tekst als toelichting.

6.1 De sterrenhemel

De standaard-subroutines worden nimmer afgedrukt. Deze zijn immers bij elk merk en type computer verschillend.

De opzet van regel 1000 is in BASICODE verplicht voorgeschreven.

```
1000 A=100:GOTO20:REM STERRENHEMEL
1010 P1=3.1415927:VR=VE
```

In de variabele `P1` wordt de waarde van `PI` gezet; de hoogte van het tekstscherm wordt in `VR` overgenomen voor later gebruik (regel 10010).

```
1020 GOSUB100:PRINT"Een momentje..."
1030 DIMM$(12),RA(110),DE(110),B(12)
1040 FORI=1 TO12:READM$(I):NEXTI
1050 FORI=1 TO110:READRA(I):NEXTI
1060 FORI=1 TO110:READDE(I):NEXTI
1070 FORI=1 TO12:READB(I):NEXTI
1080 GOSUB100
1090 SR$="ELEKTRONISCHE STERRENKAART":GOSUB150:PRINT
```

De tekst in SR$ wordt op opvallende wijze in de bovenste schermregel weergegeven.

```
1100 PRINT;PRINT"Een grafisch programma in BASICODE-3."
1110 PRINT:PRINT"Voor het verkennen van de sterrenhemel"
1120 PRINT:PRINT"worden vaak ronde sterrenkaarten"
1130 PRINT:PRINT"gebruikt. Dit programma tovert zo'n"
1140 PRINT:PRINT"kaart voor elke willekeurige datum en"
1150 PRINT:PRINT"tijd op uw scherm!"
1160 GOSUB10000
1170 PRINT"Langs de randen van de kaart is de"
```

```
1180 PRINT:PRINT"horizon aangegeven. Het midden van de"
1190 PRINT:PRINT"kaart komt overeen met het zenith, dat"
1200 PRINT:PRINT"is het punt aan de hemel dat recht"
1210 PRINT:PRINT"boven u ligt."
1220 PRINT:PRINT"Hoe verder een ster dus naar de rand"
1230 PRINT:PRINT"van de kaart staat, hoe lager hij in"
1240 PRINT:PRINT"werkelijkheid aan de hemel staat."
1250 PRINT:PRINT"Bedenk wel dat u de kaart in een keer"
1260 PRINT:PRINT"kunt overzien maar de sterrenhemel niet!"
1270 GOSUB10000
1280 PRINT"De kaart toont de sterren tot aan"
1290 PRINT:PRINT"magnitude 3, dat zijn de sterren die"
1300 PRINT:PRINT"normaal in de stad zichtbaar zijn."
1310 PRINT:PRINT"De heldere sterren zijn afgebeeld als"
1320 PRINT:PRINT"twee puntjes naast elkaar op de kaart."
```

Hier eindigt de inleiding en begint het eigenlijke programma.

```
1330 PRINT:PRINT:PRINT"Geef de dag van de maand: ";
1340 INPUTDA:IF(DA<1) OR(DA>31) THEN1330
1350 PRINT:PRINT"Geef de maand (1-12): ";
1360 INPUTMA:IF(MA<1) OR(MA>12) THEN1350
1370 PRINT:PRINT"Geef nu de tijd op (0.00-23.59): ";
1380 INPUTTY:IF(TY<0) OR(TY>24) THEN1370
```

Het programma controleert of de ingetikte datum en tijd kan kloppen. Zo niet dan wordt teruggesprongen naar de PRINT- en INPUT-opdracht.

```
1390 UR=INT(TY):MI=INT((TY-INT(TY))*100+.5)
1400 IFMI>59 THEN1370
1410 REM **TEKEN STERRENKAART**
1420 REM
1430 GOSUB600
1440 CN=0:HM=1-1/HG:RR=.2025
```

Grafisch bedrijf wordt ingeschakeld. In HM wordt berekend wat de x-coördinaat van het meest rechtse beeldpunt is; in HG staat immers het totale aantal beeldpunten en de x-waarde 1 valt volgens het protocol juist buiten het scherm.

```
1450 FORY=0 TO.45 STEP.9/VG
1460   X=SQR(RR-Y*Y)*.75
1470   HO=0:VE=Y+.5:GOSUB620:HO=.5-X:GOSUB630
1480   HO=.5+X:GOSUB620:HO=HM:GOSUB630
1490   HO=0:VE=.5-Y:GOSUB620:HO=.5-X:GOSUB630
1500   HO=.5+X:GOSUB620:HO=HM:GOSUB630
1510 NEXTY
```

Bovenstaande lus tekent met een stapgrootte die nauwelijks ongunstiger is dan wat met het aantal beeldlijnen overeenkomt een serie horizontale lijnen op het scherm waarbinnen een zwarte cirkel wordt opengelaten.

```
152Ø SR$="NOORD":HO=.4375:VE=Ø:GOSUB65Ø
153Ø SR$="ZUID":HO=.45:VE=.959:GOSUB65Ø
154Ø SR$="OOST":HO=.Ø25:VE=.5:CN=1:GOSUB65Ø
155Ø SR$="WEST":HO=.875:GOSUB65Ø
```

Op vier plaatsen wordt in het grafische scherm de windrichting geprint.

```
156Ø REM ***********************
157Ø REM **   COORDINATEN VAN   **
158Ø REM ** WAARNEMINGSPLAATS **
159Ø REM **EVENTUEEL AANPASSEN**
16ØØ REM **AAN EIGEN OMGEVING **
161Ø REM ***********************
162Ø REM
163Ø REM  GEOGRAFISCHE BREEDTE
164Ø REM     VOOR HILVERSUM
165Ø B=52.2
166Ø REM
167Ø REM  GEOGRAFISCHE LENGTE
168Ø REM     VOOR HILVERSUM
169Ø L=5.16666667
17ØØ REM
171Ø SB=SIN(B*P1/18Ø):CB=COS(B*P1/18Ø):LL=L*P1/18
172Ø ZO=1:REM TIJDZONE
173Ø IF(MA>3) AND(MA<1Ø) THENZO=2:REM ZOMERTIJD 1/4-1/1Ø
174Ø H=(UR+MI/6Ø-ZO)*P1/12
175Ø REM *DAGEN VANAF 1 JANUARI*
176Ø T=INT(275*MA/9)-2*INT((MA+9)/12)+DA-3Ø
177Ø REM
178Ø REM LOKALE STERRENTIJD
179Ø REM BEPAALT STAND VAN DE
18ØØ REM STERRENHEMEL
181Ø REM
182Ø LS=1.75Ø9289+LL+1.72Ø2765E-2*T+1.ØØ27379*H
183Ø LS=LS-2*P1*INT(LS/2/P1)
184Ø REM
185Ø REM   PLOT DE STERREN
186Ø REM
187Ø BT=1:CN=Ø
188Ø FORI=1 TO11Ø
189Ø   BR=Ø
19ØØ   IFI=B(BT) THENBR=1:BT=BT+1
191Ø   HA=LS-RA(I):SD=SIN(DE(I)):CD=COS(DE(I))
192Ø   IFHA<Ø THENHA=HA+2*P1
```

```
1930   CZ=SB*SD+CB*CD*COS(HA)
1940   IFCZ<=0 THEN2020
1950   REM STER IS BOVEN HORIZON
1960   Z1=P1/2-ATN(CZ/SQR(1-CZ*CZ)):Z=Z1*.9/P1
1970   X=Z*CD*SIN(HA)/SIN(Z1)
1980   Y=SQR(ABS(Z*Z-X*X))
1990   IFSD>SB*CZ THENY=-Y
2000   HO=X*.75+.5:VE=Y+.5
2010   GOSUB620:IFBR=1 THENHO=HO+1/HG:GOSUB620
2020 NEXTI
```

In regel 2000 worden per ster de berekende coördinaten in HO en VE gezet waarna in regel 2010 met een aanroep van subroutine 620 een punt op het scherm wordt geplot. Als de desbetreffende ster een erg heldere is (BR=1) dan wordt tevens het ernaast gelegen punt geplot om de ster ook op het scherm helderder te laten lijken.

```
2030 SR=DA:GOSUB300:SR$=SR$+" "+M$(MA)
2040 HO=.025:VE=.875:CN=1:GOSUB650
2050 SR=UR:GOSUB300:UR$=SR$+"h"
2060 SR=MI:GOSUB300:IFMI<10 THENSR$="0"+SR$
2070 SR$=UR$+SR$+"m":HO=.825:VE=.875:GOSUB650
2080 SR$=">>":HO=.9:VE=.959:CN=0:GOSUB650
2090 GOSUB210:GOSUB100:GOTO1330
```

In subroutine 300 worden diverse getallen (datum en tijd) omgezet in strings. Deze strings worden samen met enkele letters in het grafische scherm ingeplot. Daarna wordt via subroutine 210 gewacht tot de gebruiker met een toetsindruk te kennen heeft gegeven dat het programma mag verdergaan. Daarna wordt het scherm gewist en naar regel 1330 teruggesprongen.

```
10000 REM WACHT OP TOETS
10010 HO=0:VE=VR:GOSUB110:PRINT"Toets voor vervolg >";
10020 GOSUB210:GOSUB100:RETURN
```

Bovenstaande wachtroutine wordt meermalen aangeroepen.

```
25000 REM **MAANDNAMEN**
25010 DATA "jan","feb","mrt","apr","mei","jun"
25020 DATA "jul","aug","sep","okt","nov","dec"
25030 REM
25040 REM COORDINATEN VAN STERREN
25050 REM 110 X RECHTE KLIMMING
25060 REM IN RADIALEN
25070 REM
25080 DATA .03,.04,.17,.19,.3,.58,.54,.55,.6,.79
25090 DATA .82,.89,1.20,1.37,1.38,1.41,1.42,1.44,1.46,1.48
25100 DATA 1.51,1.55,1.56,1.67,1.73,1.76,1.98,2,2.03,2.47
25110 DATA 2.65,2.88,2.89,2.94,3.09,3.11,3.37,3.5,3.51,3.61
```

```
25120 DATA 3.73,3.89,4.08,4.19,4.3,4.35,4.49,4.6,4.7,4.87
25130 DATA 5.19,5.33,5.41,5.43,5.58,5.69,6.03,6.04
25140 DATA .05,.24,.37,.50,.56,.80,.97,.99,1.02,1.03
25150 DATA 1.03,1.29,1.31,1.34,1.43,1.45,1.46,1.47,1.56,1.67
25160 DATA 1.95,2.12,2.55,2.70,3.21,3.27,3.29,3.32,3.38,3.41
25170 DATA 3.64,3.80,3.86,3.88,4.00,4.12,4.21,4.25,4.29,4.32
25180 DATA 4.37,4.58,4.64,4.99,5.01,5.17,5.17,5.63,5.70,5.78
25190 DATA 5.94,3.21
25200 REM
25210 REM 110 X DECLINATIE
25220 REM IN RADIALEN
25230 REM
25240 DATA .51,1.03,.99,-.32,.62,1.56,.74,.41,-0.05,.07
25250 DATA .71,.87,.29,-.14,.8,.11,.5,-.01,-.02,-.03
25260 DATA -.17,.13,.78,-.31,.29,-.29,.56,.09,.49,-.15
25270 DATA .21,.99,1.08,.36,.26,.94,.98,.96,-.19,.86
25280 DATA .34,1.3,.47,-.39,-.46,-.18,-.27,.22,.9,.68
25290 DATA .15,.7,.79,0.59,1.09,0.17,0.49,0.26
25300 DATA .26,1.06,1.05,.36,.61,.93,.83,.42,.56,.70
25310 DATA -.24,.58,.76,-.09,-.36,-.31,-.10,.37,.65,.39
25320 DATA .15,-.42,.42,.35,-.30,-.29,-.41,-.02,.67,.19
25330 DATA .32,.67,.47,-.28,-.16,.11,-.34,-.06,1.07,.38
25340 DATA .55,.91,.08,.24,-.37,.79,.18,-.10,-.28,-.01
25350 DATA .53,1.00
25360 REM
25370 REM NUMMERS VAN DE 11
25380 REM HELDERSTE STERREN
25390 REM
25400 DATA 13,14,15,22,26,28,39,41,45,50,51,200
30000 REM *********************
30010 REM *                   *
30020 REM *   ELEKTRONISCHE   *
30030 REM *   STERRENKAART    *
30040 REM *                   *
30050 REM *********************
32000 REM *                   *
32010 REM *  JAN BREDENBEEK   *
32020 REM *DIEPENDAALSELAAN 255*
32030 REM * 1214 KD  HILVERSUM *
32040 REM *                   *
32050 REM *  OORSPRONKELIJKE  *
32060 REM *      VERSIE:      *
32072 REM *   OKTOBER 1983    *
32080 REM *                   *
32090 REM *   AANGEPAST VOOR  *
32100 REM *    BASICODE-3:    *
32110 REM *   DECEMBER 1985   *
32120 REM *                   *
```

```
32130 REM *    ZX SPECTRUM      *
32140 REM *                     *
32150 REM ***********************
```

6.2 Drie populair-klassieke stukjes

Het tweede demonstratieprogramma betreft vooral de muziekmogelijkheden in BASICODE-3. In het programma zijn die overigens niet ten volle benut: alle drie de opgenomen werken worden op een vaste sterkte gespeeld. Wie het beter kan ...

Het programma begint met het klaarzetten van enkele hulpvariabelen. In DU$(.) wordt een tabel klaargezet van alle in de stukjes voorkomende tijdsduren van de diverse tonen, in VO$(.) gebeurt hetzelfde met het volume (dus steeds ofwel 0, ofwel 9).

```
1000 A=200:GOTO20:REM muziekdemonstratie
1010 DIM DU(8),VO(8),TT$(4)
1020 DU$="032400281202060301"
1030 VO$="090900090909090009"
1040 TT$(1)="1  Preludium nr.1,      J.S.Bach"
1050 TT$(2)="2  Solfeggietto,      C.Ph.E.Bach"
1060 TT$(3)="3  Inventione nr.8,     J.S.Bach"
1070 TT$(4)="4  Geen muziek meer"
1080 SB=INT(HO-23)/2:SH=VE
1090 FOR I=0 TO 8
1100   DU(I)=VAL(MID$(DU$,2*I+1,2))/2
1110   VO(I)=VAL(MID$(VO$,2*I+1,2))
1120 NEXTI
```

Na deze opstartzaken begint het hoofdprogramma. Dat zet enkele kopregels geaccentueerd op het scherm en print de vier tekstregels die de keuzemogelijkheden aangeven.

```
1130 GOSUB100
1140 HO=SB:VE=0:GOSUB110
1150 SR$="BASICODE-3 MUZIEK":GOSUB150
1160 HO=SB-2:VE=3:GOSUB110
1170 SR$="Johann Sebastian Bach":GOSUB150
1180 VE=4:GOSUB110
1190 SR$="Carl Philip E. Bach  ":GOSUB150
1200 PRINT:PRINT:PRINT
1210 PRINT"U kunt kiezen uit:"
1220 UL=0:FOR T=1TO4
1230   GOSUB1500
1240 NEXTT
1250 PRINT:PRINT:PRINT
1260 PRINT"Welk nummer kiest u ? ";
```

Vervolgens wordt gewacht op een toetsaanslag die overeenkomt met de geboden mogelijkheden. De tekstregel van de gekozen optie wordt vervolgens geaccentueerd opnieuw geprint.

```
1270 GOSUB210:IF(IN<49)OR(IN>52) THENGOSUB250:GOTO1270
1280 T=IN-48:PRINTT:UL=1:GOSUB1500
```

Daarna wacht het programma op een druk op RETURN, waarbij iedere andere toetsindruk een nieuwe start van het hoofdprogramma oplevert.

```
1290 GOSUB210:IFIN< >13 THEN1130
1300 IFT=4 THEN950
```

Als dat ook gebeurd is, wordt de "DATA-zoek-routine" aangeroepen, meteen gevolgd door de "afspeelroutine". Daarna wordt het hoofdprogramma herstart.

```
1310 HO=0:VE=SH:GOSUB110
1320 PRINT"Even geduld, ik zoek het op";
1330 GOSUB2030
1340 GOSUB110
1350 PRINT"U hoort nu het gekozen stuk ..";
1360 GOSUB3030
1370 GOTO1130
```

De drie hulpsubroutines:

```
1470 REM
1480 REM *** REGELPRINTROUTINE ***
1490 REM
1500 HO=0:VE=8+T*2:GOSUB110
1510 SR$=TT$(T)
1520 IFUL>0 THEN GOSUB150:RETURN
1530 PRINT"    ";SR$:RETURN


2000 REM
2010 REM *** Zoek gewenste stuk in DATA-regels ***
2020 REM
2030 RESTORE
2040 IF T<=1 THENRETURN
2050 T=T-1
2060 READSP,DV:IFSP>0 THEN2060
2070 GOTO2040


3000 REM
3010 REM *** Speel het muziekstukje tot SP = 0 ***
```

```
3Ø2Ø REM
3Ø3Ø READSP,DV
3Ø4Ø SD=DU(DV):SV=VO(DV)
3Ø6Ø GOSUB4ØØ
3Ø7Ø IFSP>Ø THEN3Ø3Ø
3Ø8Ø RETURN
```

En daarna de benodigde DATA-regels, die overigens niet allemaal worden afgedrukt. U hebt ze immers ook op cassette staan. Elke te spelen noot wordt gecodeerd in twee getallen. Het eerste getal is de code (SP) voor welke toon gewenst wordt, het tweede getal is een code waaruit de benodigde combinatie voor SD en SV wordt afgelezen. De reeks wordt beëindigd met Ø,Ø.

```
25ØØØ REM  Johann Sebastian Bach
25ØØ1 REM Preludium nr.1 uit "Das Wohltemperirte Clavier"
25Ø1Ø DATA 72,Ø,76,Ø,79,Ø,84,Ø,88,Ø,79,Ø,84,Ø,88,Ø
25Ø2Ø DATA 72,Ø,76,Ø,79,Ø,84,Ø,88,Ø,79,Ø,84,Ø,88,Ø
25Ø3Ø DATA 72,Ø,74,Ø,81,Ø,86,Ø,89,Ø,81,Ø,86,Ø,89,Ø
25Ø4Ø DATA 72,Ø,74,Ø,81,Ø,86,Ø,89,Ø,81,Ø,86,Ø,89,Ø
  |
  |
2567Ø DATA 48,Ø,6Ø,Ø,65,Ø,69,Ø,72,Ø,77,Ø,72,Ø,69,Ø
2568Ø DATA 72,Ø,69,Ø,65,Ø,69,Ø,65,Ø,62,Ø,65,Ø,62,Ø
2569Ø DATA 48,Ø,59,Ø,79,Ø,83,Ø,86,Ø,89,Ø,86,Ø,83,Ø
257ØØ DATA 86,Ø,83,Ø,79,Ø,83,Ø,74,Ø,77,Ø,76,Ø,74,Ø
2571Ø DATA 48,1,Ø,2
26ØØØ REM Carl Philipp Emanuel Bach
26ØØ1 REM Solfeggietto
26Ø1Ø DATA 6Ø,Ø,48,Ø,51,Ø,55,Ø,6Ø,Ø,63,Ø,62,Ø,6Ø,Ø
26Ø2Ø DATA 59,Ø,55,Ø,59,Ø,62,Ø,67,Ø,65,Ø,63,Ø,62,Ø
26Ø3Ø DATA 63,Ø,6Ø,Ø,63,Ø,67,Ø,72,Ø,75,Ø,74,Ø,72,Ø
26Ø4Ø DATA 74,Ø,72,Ø,71,Ø,69,Ø,67,Ø,65,Ø,63,Ø,62,Ø
  |
  |
2661Ø DATA 84,Ø,86,Ø,84,Ø,83,Ø,81,Ø,79,Ø,77,Ø,75,Ø
2662Ø DATA 74,Ø,75,Ø,72,Ø,75,Ø,79,Ø,84,Ø,87,Ø,86,Ø
2663Ø DATA 84,Ø,83,Ø,79,Ø,83,Ø,86,Ø,91,Ø,89,Ø,87,Ø
2664Ø DATA 86,Ø,87,Ø,84,Ø,87,Ø,91,Ø,84,Ø,87,Ø,86,Ø
2665Ø DATA 83,Ø,84,6,Ø,2
27ØØØ REM Johann Sebastian Bach
27ØØ1 REM  Inventione nr.8
27Ø1Ø DATA 65,Ø,72,7,69,Ø,72,7,65,Ø,72,7,72,Ø,72,7
27Ø2Ø DATA 65,Ø,72,7,77,6,76,Ø,74,Ø,72,Ø,74,Ø,72,Ø
27Ø3Ø DATA 7Ø,Ø,69,Ø,7Ø,Ø,69,Ø,67,Ø,65,Ø,72,7,69,Ø
27Ø4Ø DATA 72,7,72,Ø,72,7,69,Ø,72,7,77,Ø,72,7,72,Ø
  |
  |
2745Ø DATA 65,Ø,64,Ø,62,6,67,Ø,65,Ø,64,Ø,65,Ø,64,Ø
```

```
2746Ø DATA 62,Ø,6Ø,Ø,62,Ø,6Ø,Ø,58,Ø,57,Ø,58,Ø,57,Ø
2747Ø DATA 55,Ø,53,6,65,Ø,64,Ø,65,Ø,72,7,57,Ø,72,7
2748Ø DATA 58,Ø,72,7,65,Ø,72,7,57,Ø,72,7,65,Ø,72,7
2749Ø DATA 55,Ø,72,7,64,Ø,72,7,65,4,Ø,2
3ØØØØ REM
3ØØ1Ø REM********************************
3ØØ2Ø REM
3ØØ3Ø REM   DRIE  VROLIJKE  WIJSJES
3ØØ4Ø REM   ALS  DEMONSTRATIE  VOOR
3ØØ5Ø REM   DE GELUIDSMOGELIJKHEDEN
3ØØ6Ø REM             IN
3ØØ7Ø REM   B A S I C O D E   -   3
3ØØ8Ø REM
3ØØ9Ø REM   MET  DANK AAN  DE HEREN
3Ø1ØØ REM       J.S. BACH SR.
3Ø11Ø REM       C.PH.E. BACH
3Ø12Ø REM
3Ø13Ø REM   COPYRIGHT  (C)
3Ø14Ø REM   STICHTING BASICODE
3Ø15Ø REM                1986
3Ø16Ø REM
3Ø17Ø REM********************************
```

6.3 Een adresbestand

Met het programma ADRESBESTAND kunt u van een flink aantal personen of instanties de naam, het adres, de woonplaats en de postcode invoeren, opvragen, sorteren en verbeteren. Het programma is volledig menugestuurd, goed gestructureerd en modulair van opzet. Aanpassing aan eigen wensen is daardoor redelijk eenvoudig.
Op de BASICODE-3 cassette staat na dit programma een bestandje met adressen voor computerliefhebbers. U kunt dat inlezen door in het hoofdmenu de keuze '1' op te geven, gevolgd door opnieuw de keuze '1' in het daarna komende submenu. Na inlezen van dat bestand kunnen de gegevens worden bekeken, aangevuld, veranderd en op een eigen cassette of diskette worden opgeslagen. Enig spelen met het demonstratiebestand geeft meteen een aardige indruk van de mogelijkheden van dit programma.

```
1ØØØ A=3ØØØ:GOTO2Ø:REM NAWP-BESTAND
1Ø1Ø GOSUB27Ø
1Ø2Ø MR=INT(FR/1ØØ-1Ø) : REM MAXIMUM AANTAL RECORDS
1Ø3Ø MV=5 : REM MAXIMUM AANTAL VELDEN
1Ø4Ø LR$="":FORI=1TOHO:LR$=LR$+" ":NEXTI
1Ø5Ø DIM ST(4Ø),RI$(MV),DE$(MV),BE$(MR,MV)
1Ø6Ø AR=Ø : REM Ø RECORDS AANWEZIG
1Ø7Ø VS=Ø : REM SORTEER OP NAAM
1Ø8Ø NU=1 : REM HUIDIG RECORD
1Ø9Ø READMV
11ØØ FOR J=Ø TO MV:READ DE$(J):BE$(Ø,J)="":NEXTJ
111Ø LR=VE:LK=HO
```

In regel 1010 wordt gekeken hoeveel geheugenruimte vrij is. Op grond daarvan wordt in regel 1020 berekend hoeveel records kunnen worden verwerkt als per record 100 bytes nodig zijn. Dit is overigens aan de ruime kant; in machines met weinig geheugen kan het slim zijn regel 1020 aan te passen en in MR een constante te plaatsen. Het dimensioneren vindt plaats in regel 1050, waarbij tevens enkele kleine hulparrays worden opgegeven. In Z80-computers, waar meestal de benodigde stringruimte uitdrukkelijk gedeclareerd moet worden, kan het wenselijk zijn in regel 1000 de variabele A een zo groot mogelijke waarde te geven.

```
1120 REM
1130 REM HOOFDCYCLUS
1140 REM
1150 GOSUB100
1160 SR$="A D R E S B E S T A N D":GOSUB10320
1170 PRINT:PRINT
1180 PRINT"Uw keuzemogelijkheden:"
1190 PRINT:PRINT
1200 PRINT:PRINT"1  Oud bestand inlezen"
1210 PRINT:PRINT"2  Bestand raadplegen/muteren"
1220 PRINT:PRINT"3  Bestand wegschrijven"
1230 PRINT:PRINT"4  Einde programma"
1240 MK=5:GOSUB1300
1250 GOSUB100
1260 ON KZ GOTO 1390,1700,3320,3740,3620
1270 REM
1280 REM KEUZE-OPVRAGEN-SUBROUTINE
1290 REM
1300 HO=0:VE=LR-1:GOSUB110
1310 PRINT"Wat is uw keuze      ";
1320 HO=17:VE=LR-1:GOSUB110
1330 INPUT KZ$:KZ=VAL(KZ$)
1340 IF(KZ<1)OR(KZ>MK)OR(KZ<>INT(KZ)) THEN
     GOSUB250:GOTO1300
1350 RETURN
```

De regels 1150 tot en met 1260 vormen het eigenlijke hoofdprogramma. Dit wordt gevolg door een subroutine die een keuze opvraagt. Dezelfde subroutine wordt ook verder in het programma aangeroepen. Wie het niet leuk vindt om bij een ingetikte keuze tevens op RETURN te moeten drukken, kan de INPUT-opdracht in regel 1330 vervangen door een aanroep van subroutine 210.

```
1360 REM
1370 REM MENUKEUZE 1=OUD BESTAND INLEZEN
1380 REM
1390 SR$="OUD BESTAND INLEZEN":GOSUB10320
1400 PRINT:PRINT
1410 PRINT"U kunt inlezen van:"
1420 PRINT:PRINT
```

```
1430 PRINT:PRINT"1  BASICODE-3 cassette"
1440 PRINT:PRINT"2  cassette van uw eigen computer"
1450 PRINT:PRINT"3  diskette van uw eigen computer"
1460 PRINT:PRINT"4  Toch geen bestand inlezen"
1470 MK=4:GOSUB1300
1480 IF KZ=4 THEN 1150
1490 NF=2*KZ-2:NF$="NAWP"
1500 GOSUB500  : REM OPENEN
1510 GOSUB540  : AR=VAL(IN$) :REM INVOER AANTAL RECS
1520 GOSUB540  : MV=VAL(IN$) :REM INVOER AANTAL VELDEN
1530 GOSUB540  : VS=VAL(IN$) :REM INVOER SORTEERVELD
1540 HO=0:VE=LR-1
1550 FORJ=0TOMV
1560   GOSUB540:DE$(J)=IN$
1570   FOR I=1 TO AR
1580     GOSUB110:PRINT"Veld ";:SR=J:GOSUB300
1590     PRINTSR$;", Record ";:SR=I:GOSUB300
1600     PRINTSR$;"      ";
1610     GOSUB540  : BE$(I,J)=IN$
1620   NEXTI
1630 NEXTJ
1640 GOSUB580 :REM SLUIT BESTAND
1650 NU=1
1660 GOTO1150
```

Bovenstaand programmablok leest een bestand in dat eerder is gemaakt en op cassette of diskette is geschreven. Afhankelijk van de gekozen bron wordt in regel 1490 bepaald met welke waarde voor NF de bestandsroutines 500, 540 en 580 worden aangeroepen. Tijdens het inlezen en opslaan van de gegevens zorgen de regels 1580 tot 1600 voor enige meldingen op het scherm. Na inlezen van het bestand wordt in de variabele NU de waarde 1 gezet, zodat bij bladeren in het bestand wordt begonnen bij het eerste record.

```
1670 REM
1680 REM MENUKEUZE 2 = BESTAND RAADPLEGEN/MUTEREN
1690 REM
1700 GOSUB100
1710 SR$="BESTAND RAADPLEGEN / MUTEREN":GOSUB10320
1720 PRINT:PRINT"U hebt nu ";:SR=AR:GOSUB300
1730 PRINTSR$;" records,"
1740 PRINT" gesorteerd op ";DE$(VS)
1750 PRINT:PRINT
1760 PRINT:PRINT"U kunt nu:"
1770 PRINT:PRINT"1  Bladeren in het bestand"
1780 PRINT:PRINT"2  Zoeken naar een ";DE$(VS)
1790 PRINT:PRINT"3  Een nieuw record invoeren"
1800 PRINT:PRINT"4  Bestand opnieuw sorteren"
1810 PRINT:PRINT"5  Terug naar het hoofdmenu"
```

```
182Ø MK=6:GOSUB127Ø
183Ø ON KZ GOTO 187Ø,233Ø,269Ø,297Ø,115Ø
```

Het submenu dat u krijgt na in het hoofdmenu voor '2' te kiezen, biedt vier nieuwe mogelijkheden. Wie nog geen bestand had, kan met keuze 3 zijn eigen gegevens invoeren. De verschillende keuzemogelijkheden leiden opnieuw tot programmamodulen:

```
184Ø REM
185Ø REM BLADEREN
186Ø REM
187Ø IF AR=Ø THEN GOSUB25Ø:GOTO17ØØ
188Ø IF NU<1 THEN NU=1
189Ø IF NU>AR THEN NU=AR
19ØØ GOSUB221Ø
191Ø HO=Ø:VE=LR-1:GOSUB11Ø
192Ø PRINT"(V)ooruit,  (T)erug,  (W)ijzigen,"
193Ø PRINT"(S)chrappen,                (M)enu";
194Ø GOSUB21Ø
195Ø IF(IN$="V")OR(IN$="v")THEN NU=NU+1:GOTO188Ø
196Ø IF(IN$="S")OR(IN$="s")THEN2Ø4Ø
197Ø IF(IN$="T")OR(IN$="t")THEN NU=NU-1:GOTO188Ø
198Ø IF(IN$="W")OR(IN$="w")THEN253Ø
199Ø IF(IN$="M")OR(IN$="m")THEN17ØØ
2ØØØ GOSUB25Ø:GOTO194Ø
```

Het blader-moduul biedt meer mogelijkheden dan alleen vooruit- en terugbladeren. Men kan het in beeld komende record ook wijzigen of zelfs helemaal schrappen.

```
2Ø1Ø REM
2Ø2Ø REM SCHRAPPEN
2Ø3Ø REM
2Ø4Ø HO=Ø:VE=Ø:GOSUB11Ø
2Ø5Ø SR$="DIT RECORD SCHRAPPEN":GOSUB1Ø32Ø
2Ø6Ø VE=LR-1:GOSUB11Ø
2Ø7Ø SR$="HERHAAL OF HERROEP OPDRACHT":GOSUB1Ø33Ø
2Ø8Ø GOSUB25Ø :REM PIEP
2Ø9Ø VE=LR:GOSUB11Ø
21ØØ PRINT"   (S)chrappen of (B)laderen    ";
211Ø GOSUB21Ø
212Ø IF(IN$<>"S")AND(IN$<>"s")THEN188Ø
213Ø FOR I=NU TOAR-1
214Ø FOR J=Ø TOMV:BE$(I,J)=BE$(I+1,J):NEXTJ
215Ø NEXTI
216Ø AR=AR-1
217Ø GOTO188Ø
```

Omdat schrappen nogal resoluut is, wordt bij die keuze opnieuw om een duidelijke schrapopdracht gevraagd.

```
2180 REM
2190 REM PRINT RECORD(NU)
2200 REM
2210 GOSUB100
2220 PRINT"RECORD NUMMER ";NU
2230 PRINT
2240 FOR J=0 TOMV
2250   HO=0:VE=5+3*J:GOSUB110:PRINTDE$(J)
2260   HO=10:GOSUB110:PRINT":  ";BE$(NU,J)
2270 NEXTJ
2280 RETURN
2290 GOTO1700
```

Het op het scherm weergeven van een record is in een aparte subroutine ondergebracht. Hierna komt de zoekroutine, die een bepaald gegeven opzoekt in het bestand.

```
2300 REM
2310 REM ZOEKEN
2320 REM
2330 GOSUB100
2340 PRINT"Welke ";DE$(VS);" wilt u "
2350 PRINT"     ";:INPUT Z$
2360 GOSUB2410
2370 GOTO1700
2380 REM
2390 REM ZOEKROUTINE Z$
2400 REM
2410 PRINT:PRINT"Even geduld a.u.b.";
2420 V1=1:V2=AR+1
2430 V3=INT((V1+V2)/2)
2440 PRINT".";
2450 IF BE$(V3,VS)>Z$ THEN V2=V3:GOTO2470
2460 V1=V3
2470 IF V2>V1+1 THEN2430
2480 NU=V1
2490 RETURN
```

De zoekroutine werkt volgens het 'Binary search' principe. Het zoeken is namelijk in de opbouw van het programma alleen mogelijk binnen het veld waarop het bestand gesorteerd is.

```
2500 REM
2510 REM WIJZIGEN
2520 REM
2530 GOSUB100
```

```
254Ø SR$="DIT RECORD WIJZIGEN":GOSUB1Ø32Ø
255Ø FOR J=Ø TOMV
256Ø    HO=Ø:VE=3+4*J:GOSUB11Ø
257Ø    PRINT"De ";DE$(J);" is nu:"
258Ø    PRINT"    ";BE$(NU,J)
259Ø    PRINT"Wijzigen (J/N) ";:GOSUB21Ø
26ØØ    IF(IN$< >"J")AND(IN$< >"j")THEN264Ø
261Ø    VE=VE+2:GOSUB11Ø:PRINTLR$
262Ø    GOSUB11Ø:PRINT"Nieuwe ";DE$(J);" ";
263Ø    INPUTBE$(NU,J)
264Ø NEXTJ
265Ø GOTO17ØØ
```

Bij het wijzigen van een record wordt per veld gevraagd of dat veld al dan niet veranderd moet worden.

```
266Ø REM
267Ø REM NIEUW RECORD INVOEREN
268Ø REM
269Ø GOSUB1ØØ
27ØØ IF AR=MR THEN GOSUB25Ø:GOTO17ØØ
271Ø PRINT"NIEUW RECORD INVOEREN"
272Ø FOR J=Ø TOMV
273Ø    HO=Ø:VE=7+J*3:GOSUB11Ø
274Ø    PRINTDE$(J)
275Ø    HO=11:GOSUB11Ø:PRINT":";
276Ø    INPUTRI$(J)
277Ø NEXTJ
278Ø PRINT:PRINT"Tik op A voor accoord,"
279Ø PRINT"      op C voor correctie,"
28ØØ PRINT"      op V voor "Vergeet deze invoer"";
281Ø GOSUB21Ø
282Ø IF(IN$="V")OR(IN$="v")THEN17ØØ
283Ø IF(IN$="C")OR(IN$="c")THEN269Ø
284Ø IF(IN$< >"A")AND(IN$< >"a")THEN281Ø
285Ø IFAR=Ø THENNU=1:GOTO291Ø
286Ø Z$=RI$(VS):GOSUB241Ø :REM ZOEK
287Ø NU=NU+1
288Ø FOR I=AR+1 TO NU+1 STEP -1
289Ø FOR J=Ø TOMV:BE$(I,J)=BE$(I-1,J):NEXTJ
29ØØ NEXTI
291Ø FOR J=Ø TOMV:BE$(NU,J)=RI$(J):NEXTJ
292Ø AR=AR+1
293Ø GOTO17ØØ
```

Een nieuw ingevoerd record wordt eerst opgenomen in een hulparray `RI$`. Als het record inderdaad in het bestand moet worden opgenomen, wordt eerst opgezocht waar het nieuwe record tussengevoegd dient te worden. Vervolgens wordt het op die plaats ingeschoven (regel 2870-2930).

```
2940 REM
2950 REM OPNIEUW SORTEREN
2960 REM
2970 GOSUB100
2980 PRINT:PRINT
2990 PRINT"U kunt het bestand laten sorteren op:"
3000 PRINT:PRINT
3010 FOR J=0 TOMV
3020   PRINT:PRINTJ+1;" ";DE$(J)
3030 NEXTJ
3040 MK=4:GOSUB1300
3050 VS=KZ-1:HO=0
3060 GOSUB110:PRINTLR$:VE=VE-4
3070 GOSUB110:PRINT"Ik sorteer op ";DE$(VS)
3080 PRINT"Ogenblikje ...";
3090 REM
3100 REM SORTEREN MET QUICKSORT
3110 REM
3120 V1=2:ST(1)=1:ST(21)=AR
3130 V1=V1-1:V2=ST(V1):V3=ST(V1+20)
3140 V4=V2:V5=V3:V6$=BE$(.5*(V2+V3),VS)
3150 IFBE$(V4,VS)<V6$THENV4=V4+1:GOTO3150
3160 IFBE$(V5,VS)>V6$THENV5=V5-1:GOTO3160
3170 IFV4<=V5 THENGOSUB3250:V4=V4+1:V5=V5-1
3180 IFV4<=V5 THEN3150
3190 IFV4<V3 THENST(V1)=V4:ST(V1+20)=V3:V1=V1+1
3200 V3=V5:IFV2<V3 THEN3140
3210 IFV1>1 THEN3130
3220 PRINT:PRINT"Gesorteerd op ";DE$(VS)
3230 GOSUB10030
3240 GOTO1700
3250 FOR J=0 TOMV
3260   H$=BE$(V4,J):BE$(V4,J)=BE$(V5,J):BE$(V5,J)=H$
3270 NEXTJ
3280 RETURN
```

Het hele bestand kan op elk van de aanwezige velden worden gesorteerd. Door het gebruik van 'Quicksort' verloopt dit sorteren erg snel.

```
3290 REM
3300 REM MENUKEUZE 3 = BESTAND WEGSCHRIJVEN
3310 REM
3320 SR$="BESTAND WEGSCHRIJVEN":GOSUB10320
3330 PRINT:PRINT
3340 PRINT"U kunt wegschrijven naar:"
3350 PRINT:PRINT
3360 PRINT:PRINT"1  BASICODE-3 cassette"
3370 PRINT:PRINT"2  cassette van uw eigen computer"
```

```
338Ø PRINT:PRINT"3  diskette van uw eigen computer"
339Ø PRINT:PRINT"4  Toch geen bestand wegschrijven"
34ØØ MK=4:GOSUB13ØØ
341Ø IF KZ=4 THEN 115Ø
342Ø NF=2*KZ-1:NF$="NAWP"
343Ø GOSUB5ØØ : REM OPENEN
344Ø SR=AR:GOSUB3ØØ:GOSUB56Ø
345Ø SR=MV:GOSUB3ØØ:GOSUB56Ø
346Ø SR=VS:GOSUB3ØØ:GOSUB56Ø
347Ø HO=Ø:VE=LR-1
348Ø FORJ=ØTOMV
349Ø   SR$=DE$(J):GOSUB56Ø
35ØØ   FOR I=1 TO AR
351Ø     GOSUB11Ø:PRINT"Veld ";:SR=J:GOSUB3ØØ
352Ø     PRINTSR$;", Record ";:SR=I:GOSUB3ØØ
353Ø     PRINTSR$;"     ";
354Ø     SR$=BE$(I,J) :GOSUB56Ø
355Ø   NEXTI
356Ø NEXTJ
357Ø GOSUB58Ø :REM SLUIT BESTAND
358Ø GOTO115Ø
```

Bestand wegschrijven is weer één van de modulen die vanuit het hoofdmenu worden opgeroepen. Net als bij 'bestand inlezen' wordt tijdens het schrijven het een en ander op het scherm gemeld (regels 3510 tot 3530) en heeft de gebruiker de keus uit verschillende geheugens en opslagmethoden.

```
359Ø REM
36ØØ REM MENUKEUZE 5 = NIEUW BESTAND
361Ø REM
362Ø GOSUB1ØØ
363Ø PRINT"Aantal velden (1...5)";
364Ø INPUT MV
365Ø MV=INT(MV):IF(MV<1)OR(MV>5)THENGOSUB25Ø:GOTO363Ø
366Ø MV=MV-1:FORJ=ØTOMV
367Ø   PRINT"Naam veld ";J+1;" ";
368Ø   INPUTDE$(J)
369Ø NEXTJ
37ØØ GOTO115Ø
```

Menukeuze 5 is een zogenaamde 'verborgen keuze'. Het menu meldt namelijk niet de aanwezigheid van deze mogelijkheid. De meer gevorderde gebruiker kan door middel van deze keuze het programma gebruiken voor een ander bestand dan alleen naam-adres-woonplaats-postcode. Het demonstratiebestand is door middel van deze keuze opgebouwd.

```
371Ø REM
372Ø REM MENUKEUZE 4 = EINDE
```

```
3730 REM
3740 GOSUB100
3750 PRINT:PRINT"EINDE PROGRAMMA"
3760 PRINT:PRINT"Weet u het zeker ";
3770 INPUT SR$:GOSUB330
3780 IF SR$< >"JA" THEN1150
3790 GOTO950
```

In het moduul voor 'einde programma' is een flauw grapje ingebouwd: er wordt namelijk gevraagd of men dat wel zeker wil. Op die vraag dient het antwoord 'ja' te worden ingetikt, anders wordt weer naar het hoofdmenu teruggesprongen. De toepassing van subroutine 330 in regel 3770 is wel illustratief voor het nut van die routine. Ongeacht of men 'ja' of 'JA' of 'Ja' of 'jA' intikte, het antwoord wordt als 'JA' herkend.

```
10000 REM
10010 REM WACHTROUTINE
10020 REM
10030 HO=INT((LK-30)/2):VE=LR
10040 SR$="Druk op de spatie voor vervolg"
10050 GOSUB10330
10060 GOSUB210
10070 IFIN$< >" "THEN10060
10080 RETURN
```

De wachtroutine print op de onderste schermregel een tekst en wacht vervolgens op een indruk van de spatie. Welke regel de onderste is, is in regel 1110 reeds opgeslagen in de variabele LR.

```
10090 REM
10100 REM GEACCENTUEERD CENTRAAL
10210 REM
10320 VE=0
10330 HO=0:GOSUB110
10340 FOR S=1 TOINT((LK-LEN(SR$))/2-3)
10350   PRINT" ";
10360 NEXTS
10370 GOSUB150
10380 RETURN
```

Deze laatste subroutine print de tekst die in SR$ staat gecentreerd op het scherm. In de variabele LK is immers onthouden hoe breed het scherm is, zodat de lus in 10340-10360 het juiste aantal spaties produceert. Daarna wordt SR$ geaccentueerd geprint door de aanroep van subroutine 150.

```
25000 DATA3
25010 DATA"Naam","Adres","Woonplaats","Postcode"
```

De DATA-regels bevatten het aantal velden minus 1 en de namen van de velden. Deze gegevens worden gebruikt tot een bestand wordt ingelezen of tot via de verborgen menukeuze iets anders wordt opgegeven. Let erop, dat DATA-regels steeds vanaf regelnummer 25000 dienen te komen.

```
30000 REM********************************
30010 REM
30020 REM  ADRESBESTAND IS EEN
30030 REM  DEMONSTRATIE VAN DE
30040 REM  MOGELIJKHEDEN   VAN
30050 REM  B A S I C O D E - 3
30060 REM
30070 REM
30080 REM  COPYRIGHT (C)
30090 REM  STICHTING  BASICODE
30100 REM                1986
30110 REM
30120 REM********************************
```



Appendix 1
Implementatieverschillen per computer

Implementatie-overzicht per computer

(1) = Acorn BBC
(2) = Acorn Electron
(3) = Apple II-familie
(4) = Commodore 64
(5) = Exidy
(6) = MSX-computers
(7) = P2000M
(8) = P2000T
(9) = Sinclair Spectrum
(10) = Tandy TRS-80
(11) = Spectravideo SVI-328

* = geheel afwijkend teken
D = dubbele verticale streep
G = gehele deling
H = breuk $^1/_2$
L = pijltje links
M = lang minteken
P = pond-teken
R = pijltje rechts
S = spatie
T = trema (twee puntjes bovenaan)
U = pijltje omhoog
V = breuk $^1/_4$
Z = breuk $^3/_4$

	1 AB	2 AE	3 Ap	4 C64	5 Ex	6 MSX	7 P2M	8 P2T	9 SS	10 TRS	11 SVI
tekstscherm											
regels:	25	25	24	25	30	24	24	24	24	24	24
tekens/regel:	40	40	40	40	64	40	80	40	42	40	40
kleine letters?	ja	ja	soms	ja	nee	ja	ja	ja	ja	nee	ja
karakter 64 @	.	.	.	.	*	.	.	.	.	.	.
karakter 91 [	L	.	.	.	.	.	.	L	.	.	.
karakter 92 \	H	.	.	.	.	.	H	H	.	.	.
karakter 93]	R	.	.	.	.	.	.	R	.	.	.
karakter 94 ^	U	.	.	U	.	.	.	U	.	.	.
karakter 95 –	M	.	.	L	.	.	M	M	.	.	.
karakter 96 `	P	P	.	S	.	.	P	P	P	.	.
karakter 123 {	V	.	.	*	.	.	V	V	.	.	.
karakter 124 |	D	.	.	*	.	.	.	D	.	.	.
karakter 125 }	Z	.	.	*	.	.	Z	Z	.	.	.
karakter 126 ~	G	.	.	*	.	.	T	G	.	.	.

	1 AB	2 AE	3 Ap	4 C64	5 Ex	6 MSX	7 P2M	8 P2T	9 SS	10 TRS	11 SVI
grafisch scherm											
lijnen:	256	256	192	200	90	192	72	72	192	72	192
punten/lijn:	320	320	256	288	112	192	152	78	256	80	192
tekstregels:	32	32	24	25	29	24	24	24	24	24	24
tekens/regel:	40	40	42	36	55	24	76	39	42	40	32
geluid SP											
laagste toon:	46	46	0	0	50	21	21	48	0	0	21
hoogste toon:	110	110	115	106	93	116	100	127	127	127	116

Appendix 2
Het verschil tussen BASICODE-2 en BASICODE-3

Deze paragraaf vermeldt de verschillen tussen BASICODE-2 en BASICODE-3. Een en ander is speciaal vermeld voor diegenen die vertrouwd zijn met BASICODE-2 en in het kort willen overzien wat er in BASICODE-3 is veranderd en toegevoegd. De volgorde in deze appendix komt overeen met de volgorde waarin in hoofdstuk 4 het protocol is behandeld.

A2.1 Algemene afspraken

In BASICODE-2 was het beeldscherm gedefinieerd als 24 regels van 40 tekens. De computers die meer of minder te bieden hadden, bleken in sommige programma's ietwat afwijkend te reageren.

In BASICODE-3 is de maat van 24 bij 40 slechts vermeld als redelijk gemiddelde, maar elk programma kan zich eenvoudig aanpassen aan het werkelijke beeldformaat doordat aan het begin in `HO` en `VE` de maximaal toegestane waarden voor de desbetreffende computer worden opgegeven.

Nieuw is het grafische beeldscherm dat een eigen coördinatenstelsel heeft gekregen: (0,0) in de linkerbovenhoek, (1,0) net naast de rechterbovenhoek en (0,1) net onder de linkeronderhoek. Voorts is vastgelegd dat het grafische beeldscherm in hoogte precies 3/4 van de breedte is. Wil men een cirkel tekenen dan zal het programma in feite een ellips moeten opbouwen waarvan de straal in verticale richting, uitgedrukt in de eenheden van het coördinatensysteem, 4/3 maal zo groot is als de straal in de horizontale richting. Voor het efficiënt programmeren wordt aan het begin van het programma in de variabelen `HG` en `VG` vermeld hoeveel beeldpunten er in horizontale resp. verticale richting zichtbaar zijn.

`GOTO 2Ø` (programma-aanloop)

Regel 20 bevatte in BASICODE-2 alleen een `CLEAR`-opdracht en een sprong naar regel 1010. In BASICODE-3 is daar een heleboel bijgekomen: alle variabelen wissen, scherm wissen, tekststand instellen, schermkleuren instellen (witte tekst op zwarte achtergrond), de variabelen `HO`, `VE`, `HG`, `VG` en `SV` een waarde geven en voor zover nodig de vereiste initialisaties van de subroutines uitvoeren.

`GOSUB 1ØØ` (tekstbedrijf en scherm wissen)

Deze subroutine wist niet alleen het scherm maar zorgt tevens voor de omschakeling naar tekstbedrijf (vanuit grafisch bedrijf).

GOSUB 15Ø (print opvallend)

Deze routine is nieuw toegevoegd.

GOSUB 2ØØ (kijk naar toetsenbord)

Hier is toegevoegd dat de code van de (eventueel) ingedrukte toets in de variabele IN wordt afgeleverd. Bovendien is in IN vastgelegd welke code daar komt te staan als één van de cursortoetsen of de wistoets wordt ingedrukt.

GOSUB 21Ø (wacht op toetsindruk)

Hier geldt dezelfde toevoeging als bij subroutine 200.

De volgende subroutines zijn nieuw in BASICODE-3:

GOSUB 22Ø (lees van het tekstscherm)

GOSUB 28Ø (STOP-toets uit-/aan-schakelen)

GOSUB 33Ø (hoofdletters maken)

GOSUB 4ØØ (muziek)

GOSUB 45Ø (afbreekbare wachtroutine)

GOSUB 5ØØ (bestand openen)

GOSUB 54Ø (invoer uit bestand)

GOSUB 56Ø (uitvoer naar bestand)

GOSUB 58Ø (bestand afsluiten)

GOSUB 6ØØ (grafisch bedrijf en scherm wissen)

GOSUB 62Ø (plot 1 punt)

GOSUB 63Ø (trek een lijnstuk)

GOSUB 65Ø (tekst op het grafische scherm)

GOTO 95Ø (Programma-beëindiging)

De sprong naar deze regel is nieuw en komt in de plaats van de in BASICODE-3 verboden geworden opdrachten END en STOP. Elk BASICODE-3 programma dient te eindigen met een sprong naar deze regel.

GOTO 1ØØØ (Programmastart)

De sprong naar deze regel is niet echt nieuw, maar is van meer betekenis geworden omdat in BASICODE-3 de opdracht RUN verboden is.

A2.2 Toegestane BASIC-opdrachten

Ook hier worden alleen de verschillen gemeld ten opzichte van BASICODE-2. Voor een volledige beschrijving zij verwezen naar hoofdstuk 4.

Toegestane BASIC-opdrachten:

ABS	AND	ASC	ATN	CHR$	COS	DATA	DIM
EXP	FOR	GOSUB	GOTO	IF	INPUT	INT	LEFT$
LEN	LET	LOG	MID$	NEXT	NOT	ON	OR
PRINT	READ	REM	RESTORE	RETURN	RIGHT$	SGN	SIN
SQR	STEP	TAB	TAN	THEN	TO	VAL	

AND, OR en NOT (Logische operatoren). Het gebruik van haakjes om de bewerkingsvolgorde duidelijk aan te geven is in BASICODE-3 verplicht (in BASICODE-2 alleen aanbevolen).

END, RUN en STOP zijn in BASICODE-3 verboden.

A2.3 De opbouw van een BASICODE-3 programma

De nu volgende indeling voor de regelnummers was in BASICODE-2 alleen een dringende aanbeveling, in BASICODE-3 is de indeling verplicht.

0 - 999: De standaardroutines.
1000: De eerste regel van het BASICODE-3 programma. Deze is verplicht van de volgende vorm:

1000 A=<waarde>:GOTO 20:REM programmanaam

1010 - 19999: het hoofdprogramma
20000 - 24999: onvermijdelijke subroutines met verboden opdrachten.
25000 - 29999: eventueel benodigde DATA-regels.
30000 - 31999: REM-regels met een korte beschrijving.
32000 - 32767: REM-regels met de naam van de auteur.

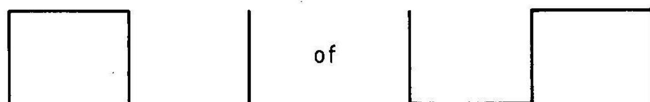
Appendix 3
Wat schrijft BASICODE op de cassette?

In dit aanhangsel wordt beschreven hoe programma's en bestanden op de cassette worden geschreven. Er is gebruik gemaakt van een structuur die bestaat uit lagen. Iedere laag maakt gebruik van wat er in de onderliggende laag is gedefiniëerd. De beschrijving zal van onderaf gemaakt worden. Dan is het nooit zo dat pas verderop duidelijk wordt wat er precies bedoeld wordt.

A3.1 Laag 1, over nullen en enen

In de eerste laag is vastgelegd hoe nullen en enen op de cassette worden gezet. Voor het schrijven van BASICODE wordt vrijwel altijd gebruik gemaakt van een gewone cassetterecorder. Het BASICODE-schrijfprogramma maakt blokspanningen, die worden toegevoerd aan de microfoongang van de recorder. Deze moet dan in de stand 'opnemen' staan. De betrouwbaarheid van de opname is groter wanneer er geen erg hoge en geen erg lage tonen nodig zijn. Hieraan is bij BASICODE voldaan.

Een 0 wordt geschreven als een volle periode van 1200 Hz:



Een 1 wordt geschreven als twee volle perioden van 2400 Hz:



Een 1 duurt dus altijd even lang als een 0. Er mag geen ruimte zitten tussen de nullen en enen. Zijn we dus eenmaal begonnen met een 0 volgens het linkse tekeningetje, dan moeten alle volgende nullen en enen volgens de linkse tekeningetjes worden gemaakt. Maar het mag ook allemaal volgens de rechtse tekeningetjes.

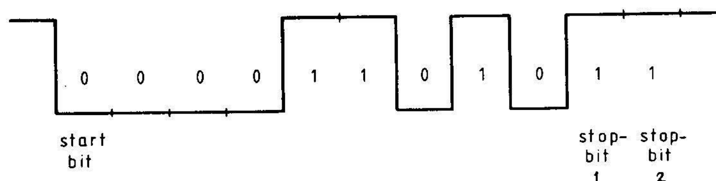
Met deze eerste laag kan dus een lange rij nullen en enen worden opgenomen en weer afgespeeld. De seinsnelheid is daarbij precies 1200 baud (1200 bits per seconde).

A3.2 Laag 2, van bits naar bytes

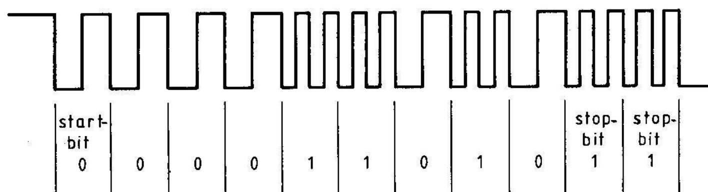
Een byte (8 bits) wordt serieel, dat wil zeggen met alle bits na elkaar, overgedragen.

Opbouw: 1 startbit (logische 0)
8 databits, het minstwaardige voorop
2 stopbits (logische 1)

Als voorbeeld gebruiken wij het byte hex 58. In nullen en enen achter elkaar gezet, ziet dit er als volgt uit:



Verwerken we ook de eerste laag, dan blijkt dat er op de cassette geschreven wordt:



(Startbit - hex 58 - twee stopbits, dus 0 0 0 0 1 1 0 1 0 1 1)
(tekening A3.4)

De tweede laag geeft dus aan hoe steeds 8 bits worden verpakt tot een 'pakketje' met een begin en een einde. Laag 1 en laag 2 samen beschrijven hoe een lange rij bytes moeten worden opgenomen op de cassette.

A3.3 De checksum

Bij het lezen van programma's en bestanden kunnen fouten optreden. Meestal gebeurt dit doordat de cassetteband beschadigd is of, bij radio, doordat er storing in de ontvangst optreedt. In zo'n geval worden er tekens verminkt. BASICODE heeft geen mogelijkheid de verminkingen zelf te herstellen. Wel kan worden herkend dat er iets

fout is gegaan. Dit gebeurt door een checksum mee te sturen die behoort bij een aantal bytes.

Deze checksum wordt berekend door de bits-gewijze exclusive-or uit te voeren over de desbetreffende bytes. Het aldus verkregen byte (8 bits) is de checksum.

```
Voorbeeld:   hex   binair
              82 = 1ØØØ ØØ1Ø
              C7 = 11ØØ Ø111
              8D = 1ØØØ 11Ø1
              83 = 1ØØØ ØØ11
             ---------------- XOR
     checksum  = Ø1ØØ 1Ø11 = 4B hex
```

A3.4 Laag 3, een programmablok

Een BASICODE-programma wordt als een blok weggeschreven. Het aantal bytes in het blok wordt bepaald door de lengte van het programma. Een compleet programma wordt als volgt geschreven:

Aanlooptoon: 5 seconden lang stopbits (een toon van 2400 Hz)
Startbyte: hex 82 (= ASCII-code 'Start of text')
Programma-bytes, het meest significante bit geïnverteerd
Stopbyte: hex 83 (= ASCII-code 'End of text')
Checksum: over het voorgaande geheel, startbyte t/m stopbyte
Uitlooptoon: 1 seconde lang stopbits (een toon van 2400 Hz)

Aan de aanlooptoon kan het BASICODE-leesprogramma merken dat er een programma gaat komen. Het wacht dan op het byte 82. Daarna worden de programmabytes gelezen totdat het byte 83 komt. Een byte met waarde hex 83 komt niet voor in het programma zelf. Een controle van de checksum geeft aan of het programma foutloos is ontvangen. Laag 3 stelt ons in staat een BASICODE-programma als geheel te herkennen en te controleren.

A3.5 Laag 3, bestandsblokken

In bestanden is de derde laag iets anders opgebouwd. Dit is nodig om een onderscheid te maken tussen programma's en bestanden. Er worden nu blokken van een vaste lengte gebruikt. Bij een bestand dat niet in één blok past, worden automatisch volgende blokken gebruikt.

Eerste blok:
Aanlooptoon: 5 seconden lang stopbits (een toon van 2400 Hz)
Startbyte: hex 81 (= ASCII-code 'Start of header')
Bloknummer: hex 80 (bloknummer = 00 met geïnverteerd msb)
Eerste 1024 bestands-bytes, het meest significante bit geïnverteerd
Stopbyte: hex 83 (= ASCII-code 'End of text')
Checksum: over de voorgaande 1027 bytes
Uitlooptoon: 1 seconde lang stopbits (een toon van 2400 Hz)

Volgende blokken:
Aanlooptoon: 5 seconden lang stopbits (een toon van 2400 Hz)
Startbyte: hex 81 (= ASCII-code 'Start of header')
Bloknummer: hex 81 (82 enz.) (bloknummer met geïnverteerd msb)
Volgende 1024 bestands-bytes, het meest significante bit geïnverteerd
Stopbyte: hex 83 (= ASCII-code 'End of text')
Checksum: over de voorgaande 1027 bytes
Uitlooptoon: 1 seconde lang stopbits (een toon van 2400 Hz)

Laatste blok:
Aanlooptoon: 5 seconden lang stopbits (een toon van 2400 Hz)
Startbyte: hex 81 (= ASCII-code 'Start of header')
Bloknummer: hex xx (bloknummer met geïnverteerd msb)
Resterende bytes, het meest significante byte geïnverteerd
Stopbyte: hex 84 (= ASCII-code 'End of tape')
Dummybytes zonder betekenis, zodat totaal aantal 1027 wordt
Checksum: over de voorgaande 1027 bytes
Uitlooptoon: 1 seconde lang stopbits (een toon van 2400 Hz)
Laag 3 stelt ons in staat data op de cassette te zetten. Er mogen geen bytes met waarde hex 84 in voorkomen.

A3.6 Laag 4, programmacodering

Een programma in BASICODE wordt opgeslagen als een tekstfile. Hierin staat het programma in de vorm waarin het als listing op het scherm of op de printer verschijnt. Toegestaan zijn alleen ASCII-waarden van Hex 20 t/m Hex 7E (dus geen 7F!). Tevens komt de 0D (= ASCII-code 'carriage return') voor als scheiding tussen de programmaregels. De laatste programmaregel moet ook zijn afgesloten met een 'carriage return'.

A3.7 Laag 4, bestandencodering

In de bestanden van BASICODE-3 kunnen uitsluitend strings worden weggeschreven. Numerieke gegevens worden door het toepassingsprogramma eerst tot strings gevormd. Alle strings worden afgesloten met een 'carriage return' (Hex 0D). De strings staan achter elkaar in het bestand. Door de indeling in blokken, veroorzaakt door de derde laag, kan een string in tweeën geknipt worden. Dit is ook de bedoeling.

A3.8 Enkele kanttekeningen

Polariteiten

Laag 1 laat toe dat BASICODE in twee polariteiten kan voorkomen. Dat is ook nodig, want er is geen vast verband in polariteit tussen het ingangs- en uitgangssignaal van cassetterecorders en andere audio-apparatuur. Het betekent wel dat een goed BASICODE leesprogramma beide flanken van het signaal gebruikt, en zelf de meest gunstige kiest. Wordt er maar naar één soort flanken gekeken, dan blijkt dat het lezen van BASICODE soms probleemloos gaat en soms helemaal niet, afhankelijk van de gebruikte apparatuur.

Blokgolven en boventonen

Het BASICODE-signaal lijkt in eerste instantie slechts opgebouwd uit de tonen 1200 Hz en 2400 Hz. Omdat echter geen sinusvormige signalen maar blokspanningen worden aangeboden aan de cassetterecorder is een groot aantal boventonen aanwezig. Het blijkt dat deze rijkdom aan hoge frequenties soms roet in het eten gooit. Speciaal zgn. HIFI-recorders doen zoveel moeite de hoge frequenties weer op te peppen dat daardoor de voor BASICODE noodzakelijke golfvormen sterk vervormd worden. Een eenvoudige rechttoe, rechtaan monorecorder voldoet meestal het beste. Wanneer een toonregeling aanwezig is, kan het nuttig blijken hiermee wat te experimenteren. Hoge tonen en lage tonen weg- of opdraaien helpt vaak in lastige gevallen.

Minimale frequentieband

De minimaal benodigde frequentieband voor BASICODE-overdracht loopt van 600 Hz tot 3000 Hz. Een filter dat alleen dit gebied doorlaat, verwijdert alle overbodige stoorsignalen. Let er echter op dat zo'n filter niet te veel fasevervorming introduceert. Het BASICODE-signaal kan daar niet tegen. Hoe scherper de filters, hoe groter de kans dat er ongewilde fasevervorming optreedt.

Inverteren van bit 7

Bij het schrijven van BASICODE wordt het meest significante bit geïnverteerd. Hierdoor is het bijna altijd zo, dat dit bit als een 1 op de cassette geschreven wordt. Dit is gedaan omdat bit 7 het laatste databit is, het komt direct voor de twee stopbits. Wanneer door een of andere oorzaak er 'een gat in het signaal valt', moet het leesprogramma opnieuw synchroniseren op de startbits. Het door de inversie van bit 7 ontstane grote blok van drie 'stopbits' zorgt ervoor dat de synchronisatie snel hersteld is. Het aantal verminkte tekens wordt daardoor beduidend minder.

Bit 7 van de checksum

Door de gekozen vorm van de bytes in programma's en bestanden is het steeds zo dat het meest significante bit = 1 is. Dit geldt ook voor de startbytes en stopbytes. De checksum is echter een bits-gewijze exclusive-or operatie die identiek werkt voor alle acht bits van een byte. Bij programma's is in de helft van de gevallen bit 7 van de checksum dus 0. (Bij bestanden is dit bit toevallig steeds 1, omdat het aantal voorafgaande bytes oneven is.)

Appendix 4
De in BASICODE-3 toegestane ASCII-codes

13 'carriage return'; begin op de volgende regel
28 cursor links
29 cursor rechts
3Ø cursor omlaag
31 cursor omhoog
127 wis het vorige karakter

32		48	Ø	64	@ *	8Ø	P	96	` *	112	p
33	!	49	1	65	A	81	Q	97	a	113	q
34	"	5Ø	2	66	B	82	R	98	b	114	r
35	#	51	3	67	C	83	S	99	c	115	s
36	$	52	4	68	D	84	T	1ØØ	d	116	t
37	%	53	5	69	E	85	U	1Ø1	e	117	u
38	&	54	6	7Ø	F	86	V	1Ø2	f	118	v
39	'	55	7	71	G	87	W	1Ø3	g	119	w
4Ø	(	56	8	72	H	88	X	1Ø4	h	12Ø	x
41	)	57	9	73	I	89	Y	1Ø5	i	121	y
42	*	58	:	74	J	9Ø	Z	1Ø6	j	122	z
43	+	59	;	75	K	91	[	1Ø7	k	123	{ *
44	,	6Ø	<	76	L	92	\ *	1Ø8	l	124	| *
45	-	61	=	77	M	93	]	1Ø9	m	125	} *
46	.	62	>	78	N	94	^	11Ø	n	126	~ *
47	/	63	?	79	O	95	_ *	111	o		

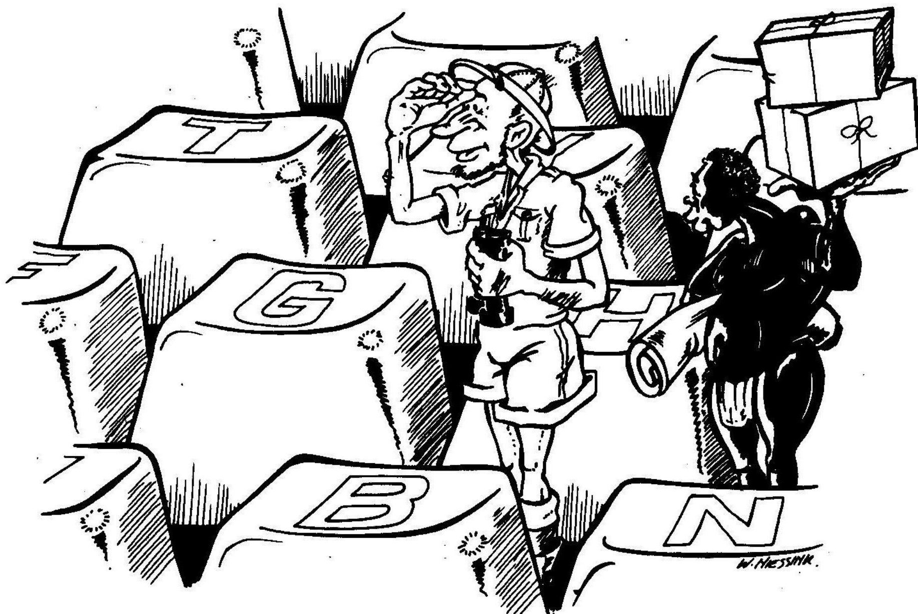
N.B.1: De cursorbesturingscodes en de code voor 'wis het vorige karakter' zijn alleen van toepassing bij het ontvangen van karakters via de subroutines 200, 210 en 450. Het is niet toegestaan, één van deze codes te gebruiken in de vorm: `PRINT CHR$(28)`. Cursorbesturing is in BASICODE-3 alleen toegestaan door middel van het gebruik van subroutine 110.

N.B.2: De ASCII-codes 96 t/m 126 zijn in het gebruik aan enige beperkingen onderhevig. Sommige subroutines (de subroutines 200,210,...) geven nimmer een van deze codes als antwoord in de variabele `IN`.

N.B.3: Met name in de Commodore-computers wordt intern op een afwijkende wijze omgegaan met deze ASCII-codes. Een opdracht als `PRINT CHR$(65)` geeft daar niet de hoofdletter A maar de kleine letter a. Wil men de hoofdletter A zien dan gebruike

men PRINT"A". In die computers zal ASC("A") de waarde 193 opleveren (128+65). Om die reden wordt het gebruik van de functies ASC en CHR$ in BASICODE-3 ontraden.

N.B.4: De codes die van een * zijn voorzien, geven in ten minste één type computer een afwijkend teken. Zie daarvoor appendix 1.



Appendix 5
Gebruikersverenigingen en andere adressen

In de opzet van dit boek mag een overzichtje van adressen niet ontbreken, vooral van díe adressen die voor de BASICODE-gebruiker van belang kunnen zijn. Helaas zijn dergelijke adressen nog wel eens aan verandering onderhevig.

Nu in BASICODE-3 de mogelijkheid van bestanden beschikbaar is, worden de diverse adressen dan ook niet in de tekst van dit boek vermeld, maar op de bijgaande cassette.

Wilt u de adresgegevens inzien?

- Lees dan eerst het BASICODE-3 vertaalprogramma voor uw eigen computer in. Hoe dat moet, hebt u eerder in dit boek kunnen vinden.
- Start dat vertaalprogramma.
- Zoek dan op de bijgaande cassette het begin op van het programma `"Adresbestand"`, dat in BASICODE-3 formaat is opgenomen.
- Lees dat programma in met behulp van het reeds ingelezen vertaalprogramma.
- `RUN` vervolgens het programma `"Adresbestand"` en kies in het openingsmenu voor "Inlezen oud bestand". Het desbetreffende bestand met namen en adressen staat meteen na het bestandsprogramma op de cassette.
- Kies na het inlezen van het bestand voor "Bladeren" en de adressen komen één voor één op uw scherm.
- U kunt desgewenst zelf het adressenbestand laten wegschrijven op een eigen cassette of diskette. U kunt ook de adresgegevens uitbreiden of verbeteren, en zodoende permanent over een compleet overzicht beschikken van alle adressen op computergebied die voor u van belang kunnen zijn.

Veel genoegen met deze toepassing van BASICODE-3.

Appendix 6
Over auteursrechten

De tekst van een boek is altijd door iemand geschreven. Datzelfde geldt voor teksten van liedjes, muziek en toneelstukken en datzelfde geldt ook voor computerprogramma's. Wie iets maakt, is de eigenaar van wat hij gemaakt heeft. Hij mag daar mee doen wat hij wil, het geheim houden, het verkopen, het vernietigen of het weggeven. Deze eigendomsrechten worden auteursrechten genoemd. Voor boeken en stukjes in tijdschriften is het meestal zó dat een uitgever de auteursrechten koopt van de auteur. De uitgever zorgt er dan voor dat het gedrukt wordt en dat de boeken te koop zijn in de boekwinkel.

Wie een boek heeft gekocht, mag daar niet zomaar mee doen wat hij wil. Want je wordt wel eigenaar van het boek, dat wil zeggen het papier, de drukinkt en de omslag, maar dat geldt niet voor de inhoud. Een boek wordt verkocht met een bepaalde bedoeling, en alleen daarvoor mag het gebruikt worden. Je hebt het om te lezen, eventueel om er stukjes uit voor te lezen en je mag het in je eigen boekenkast zetten. Je mag het ook verbranden maar je mag het niet tegen betaling of in onbeperkte mate uitlenen en, wat zeker niet mag, het in zijn geheel als hoorspel voor de radio brengen. Dat kan alleen als daarvoor eerst toestemming is gegeven door de auteur van het werk.

Zo is het ook met computerprogramma's. Alle programma's in deze uitgave zijn van iemand. Van wie? Kijk maar eens in de listing. Daarin staat meestal wel de naam van de man of vrouw die het gemaakt heeft. Voor deze uitgave van BASICODE-3 heeft Kluwer Technische Boeken B.V. toestemming verkregen om het BASICODE-3 protocol, de vertaalprogramma's en de begeleidende teksten gebundeld uit te geven in de vorm van een boek en een cassette. Wanneer de programma's worden uitgezonden op de radio, dan heeft de desbetreffende omroep daar apart toestemming voor gekregen.

U hebt nu deze uitgave gekocht. Daarmee hebt u het recht verworven op het normale gebruik ervan. Een fotokopie maken van enkele pagina's uit het boek is alleen toegestaan als dat voor uw eigen gebruik is. Datzelfde geldt ook voor de computerprogramma's. Voor uzelf mag u best enkele kopieën maken van het vertaalprogramma voor uw merk computer. Dat is veel handiger in het gebruik dan het steeds weer tussen de aankondigingen uit te moeten vissen op de originele cassette. Maar u mag geen kopie weggeven of verkopen aan iemand anders.

In het geval van BASICODE-3 is het zó geregeld dat de auteurs eigenaar blijven van hun eigen vertaalprogramma. Zij hebben het recht, hun eigen vertaalprogramma dus

wel verder te verspreiden, los van deze uitgave. Komt u dus een BASICODE-3 vertaalprogramma tegen in het programma-aanbod van de gebruikersvereniging van uw computermerk, dan kan dat alleen als de auteur van dat programma het zelf heeft aangeboden aan de club. Als hij daarbij heeft gezegd dat iedereen er gratis gebruik van mag maken, dan kan dat. Maar dat geldt dan alleen voor dát vertaalprogramma, en alleen als u het van de vereniging krijgt.

Bij programma's geschreven in BASICODE, ligt het al net zo. Ook zo'n programma is van iemand, ook al hebt u het bijvoorbeeld opgenomen van de radio of ingetikt van een listing in een tijdschrift. U mag het programma alleen zelf gebruiken. Als u er de naam van de auteur uit haalt, dan verandert daar niets aan. Zelfs als u hier en daar wat wijzigt, dan hebt u daarmee nog geen eigen programma gemaakt. Daarvoor moet de opzet grondig anders zijn en ook de uitvoering moet duidelijk verschillen. Hier geldt ook weer: alleen voor eigen gebruik mag een kopie gemaakt worden.

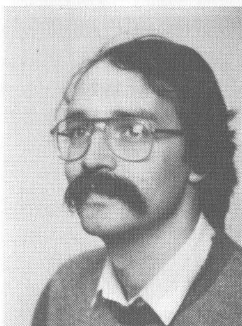
Het lijkt moeilijk te worden wanneer het gaat om zelf geschreven programma's in BASICODE. Hierin wordt immers gebruik gemaakt van de standaardroutines van BASICODE. Deze moeten deel uitmaken van het programma wil het naar behoren werken. Toch is dit geen probleem. Het kernpunt van BASICODE is de overdraagbaarheid naar andere merken. Wanneer daartoe het programma op de BASICODE-manier op de cassette wordt gezet, worden de standaardroutines eerst verwijderd. Een ander merk of type computer heeft andere routines nodig. Op dat moment is het hele programma voor honderd procent eigen werk. Niemand hoeft zich dus geremd te voelen om zijn eigen programma, geschreven volgens de regels van BASICODE, te beschouwen als eigen werk. Eigen werk is eigen werk, zelfs als dat eigen werk alleen werkt in combinatie met de standaardroutines van BASICODE. Ook hier geldt dat niemand zonder toestemming van de maker dat programma mag verkopen of verspreiden of wat dan ook.

Dan is er nog een belangrijk punt. De naam 'BASICODE-3' is verbonden met de inhoud van deze uitgave. Programma's in BASICODE-3 werken alleen dán in alle computers goed, als alle regels van het protocol nauwkeurig worden opgevolgd. Wie daar wat slordig in is, maakt een programma dat wel op BASICODE lijkt, maar het in feite niet is. Het zal vast wel werken op de eigen computer en misschien ook wel op een bepaald ander merk. Maar als het wordt geprobeerd op alle merken, komen ineens de problemen aan het licht. Daarom is het erg belangrijk zorgvuldig volgens de regels te werken. Want alleen als dat allemaal goed gedaan is, dan alleen mag u het noemen: een programma in BASICODE-3.

Appendix 7
De mensen achter BASICODE



Klaas Robers, PA0KLS, (1944). Ingenieur bij Philips te Eindhoven. Uitvinder van het systeem en de naam BASICODE. Voorzitter van de Stichting BASICODE. Stimulerende kracht en medeontwikkelaar van BASICODE-2 en BASICODE-3.

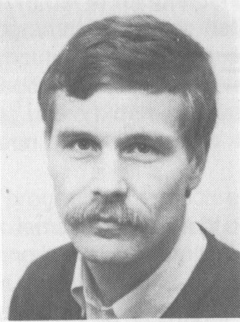


Jochem Herrmann, PE0JHM, (1956). Werkzaam als ontwikkelaar bij een klein 'High-Tech' bedrijf op het gebied van Robotvision. Maakte voor de APPLE de vertaalprogramma's voor BASICODE-1, -2 en -3. Secretaris van de Stichting BASICODE en medeontwikkelaar van het BASICODE-2 en BASICODE-3 protocol.



Jacques Haubrich (1942). Leraar wiskunde en informatica te Eindhoven. Sinds 1961 met computers bezig. Maakte voor BASICODE-1, -2 en -3 vertaalprogramma's voor diverse Commodore-computers (PET, CBM 3000-, 4000- en 8000-series, VIC-20, C-64). Penningmeester van de Stichting BASICODE en medeontwikkelaar van het BASICODE-2 en BASICODE-3 protocol.

CMG



Peter Maathuis (1951). Werkt bij een groot software-house als adviseur op het gebied van datacommunicatie. Maakte na de aanschaf van een van de eerste BBC-computers meteen een BASICODE inleesprogramma. Hieruit groeiden de vertaalprogramma's voor BASICODE-2 en BASICODE-3. Hij is daarnaast aktief als bestuurslid van de Stichting BASICODE.



Harm Mulder (1957). Electronica ingenieur THT, werkzaam op het Dr. Neher-lab van de PTT. Tijdens zijn studietijd begonnen met de computerhobby met een MicroProfessor. Sinds 1984 in bezit van een Acorn Electron. Maakte daarvoor na veel puzzelwerk de BASICODE-2 en BASICODE-3 vertaalprogramma's.



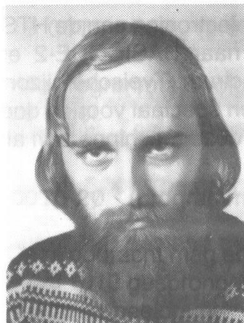
Hermine Bakker (1921). Huisvrouw en computerfanate. Leverde een belangrijk aandeel in de vertaalprogramma's voor BASICODE-2 en BASICODE-3 voor de Exidy Sorcerer computer.



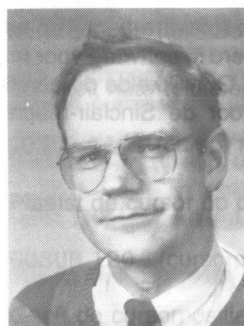
Henk Warnitz, PE1LIO, (1951). Electronicus, computerhobbyist, zendamateur en muziekliefhebber. Ontwerper van de bij de Exidy Sorcerer benodigde hardware-interfaces en mede-auteur van de BASICODE-2 en -3 vertaalprogramma's voor deze machine.



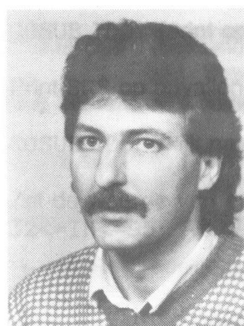
Theo van Lieshout, (1953). Electronicus bij het Van der Waals laboratorium van de Universiteit van Amsterdam. Maakte voor de DAI-computer de vertaalprogramma's voor BASICODE-1 en BASICODE-2. Was destijds één van de eersten die het idee van een 'BASICODE' deed.



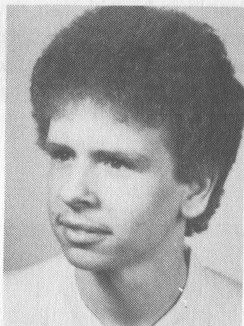
Peter Zevenhoven, (1956). Ontwerper, programmeur en (vooral) computerhobbyist. Mede-oprichter van de Spectravideo Computer Users Club. Auteur van het BASICODE-2 en BASICODE-3 vertaalprogramma voor de SV.328 en MSX-computers.



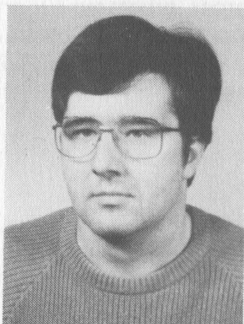
Henk Koorn (1949). Werkzaam als groepsleider bij een groot elektronica-bedrijf. Hobby's: muziek, elektronica en computers. Ontwikkelde voor BASICODE-2 en BASICODE-3 het vertaalprogramma voor de P2000M op basis van het programma voor de Exidy Sorcerer.



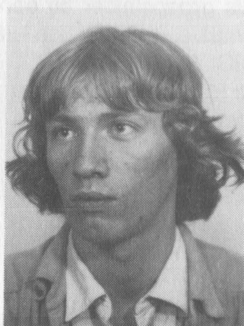
Frans van Dal (1954). Werkt bij een bekend concern in het zuiden des lands als ontwikkelaar van VMEbus/68000 produkten. Hij maakte de BASICODE-1 en BASICODE-2 vertaalprogramma's voor de P2000. Zijn hobby's zijn, naast computers, tennis, motorrijden, koken en sportief kamperen.



Yoeri Geutskens (1970). Scholier, vierde klas Atheneum. Sinds 1981 in zijn schaarse vrije tijd met de P2000T bezig. Bouwde het bestaande BASICODE-2 vertaalprogramma voor de P2000T uit en om tot het op de cassette opgenomen BASICODE-3 vertaalprogramma voor deze machine.



Jan Bredenbeek (1964). Student electronica aan de HTS. Ontwikkelde vertaalprogramma's naar BASICODE-2 en BASICODE-3 voor de Sinclair Spectrum. Typische bijzonderheid van die programma's is een speciaal voor dit doel ontwikkelde BASIC, waardoor ook deze machines zich als echte computers gedragen.



Rik Koevoets (1960). Werkt bij een software-bedrijf op het gebied van videotext. Gespecialiseerd in software voor minimale hardware-omstandigheden. Ontwikkelde de eerste BASICODE-vertaalprogramma's voor de Sinclair-machines ZX-81 en Spectrum.

Appendix 8
Het BASICODE-3 protocol in het kort

A8.1 De BASICODE-3 standaardroutines

`10 GOTO 1000` (programma-start)

Alle programma's volgens het BASICODE-3 protocol beginnen (automatisch) met deze regel.

`GOTO 20` (programma-aanloop)

Deze opdracht mag en moet alleen in regel 1000 staan. Vanuit deze regel wordt naar regel 1010 gesprongen met in `HO,VE` de maten van het tekstscherm en in `HG,VG` de aantallen beeldpunten van het grafische scherm.

`GOSUB 100` (tekstbedrijf en scherm wissen)

Schakelt naar tekstbedrijf, wist het scherm en plaatst de cursor op de positie 0,0.

`GOSUB 110` (cursor naar `HO,VE`)

Plaatst de cursor op positie `HO,VE` op het tekstscherm.

`GOSUB 120` (cursorpositie in `HO,VE`)

Geeft de cursorpositie in `HO` en `VE`.

`GOSUB 150` (print opvallend)

Print `SR$` op opvallende wijze, vooraf en erna drie spaties.

`GOSUB 200` (kijk naar toetsenbord)

Zet de waarde van een ingedrukte toets in `IN$` en de code ervan in `IN` (normaal: `32<=IN<=95`). Als geen toets is ingedrukt dan `IN$=""` en `IN=0`.

Speciale codes (decimaal):
nieuwe regel : 13
cursor links : 28
cursor rechts : 29
cursor omlaag : 30
cursor omhoog : 31
wis/delete : 127

GOSUB 21Ø (wacht op toetsindruk)

Wacht tot een toets is ingedrukt en zet de waarde van deze toets in de variabelen IN$ en IN. Zie subroutine 200.

GOSUB 22Ø (lees van het tekstscherm)

Geeft in IN de code van het karakter dat op positie HO,VE op het scherm zichtbaar is (32<=IN<=95).

GOSUB 25Ø (attentiesein)

Produceert een attentiepiepje.

GOSUB 26Ø (geef randomgetal)

Geeft een randomgetal in de variabele RV (Ø<=RV<1).

GOSUB 27Ø (meld vrije ruimte)

Geeft aantal vrije bytes in FR.

GOSUB 28Ø (STOP-toets schakelen)

FR=1 dan stoptoets uitschakelen.
FR=Ø dan stoptoets inschakelen.

GOSUB 3ØØ (omzetting in string)

Zet SR om in SR$ zonder spaties aan begin of eind.

GOSUB 31Ø (getal formatteren)

Zet SR om in SR$ met lengte CT en CN decimalen.

GOSUB 33Ø (hoofdletters maken)

Verandert alle in SR$ aanwezige kleine letters in hoofdletters.

GOSUB 35Ø (tekst naar printer)

Print SR$ op de printer, sluit de regel niet af.

GOSUB 36Ø (regelopvoer printer)

Sluit de printerregel af en begint op een nieuwe printerregel.

GOSUB 4ØØ (muziek)

Produceert een toon met toonhoogte SP, duur SD en volume SV.
Ø<=SP<=127. SD in 0,1 seconden, Ø<=SV<=15.

GOSUB 45Ø (afbreekbare wachtroutine)

SD is een wachttijd in 0,1 seconden. Beëindigt wachten na opgegeven tijd of als toets is ingedrukt. Toets afleveren in IN$ en IN op manier van subroutine 200.

GOSUB 5ØØ (bestand openen)

Opent een bestand met naam NF$ en code NF.
NF=0 of 1 : BASICODE-bestand op cassette
NF=2 of 3 : bestand naar standaard-geheugen
NF=4 of 5 : bestand op diskette
NF=6 of 7 : bestand op diskette

even nummers = invoer
oneven nummers = uitvoer

GOSUB 54Ø (invoer uit bestand)

Leest IN$ uit bestand NF met controlecode in IN. Alles in orde als IN=Ø.

GOSUB 56Ø (uitvoer naar bestand)

Voert SR$ uit naar bestand met controlecode NF. Na afloop is IN=Ø als alles in orde is.

GOSUB 58Ø (bestand afsluiten)

Sluit bestand met code NF.

GOSUB 6ØØ (grafisch bedrijf en scherm wissen)

Schakelt naar grafisch bedrijf en wist het scherm.

GOSUB 62Ø (plot 1 punt)

Zet de grafische cursor op het punt HO,VE en plot dat punt. CN=Ø geeft voorgrondkleur, CN=1 geeft achtergrondkleur. (HO en VE minimaal 0 en altijd kleiner dan 1).

GOSUB 63Ø (trek een lijnstuk)

Trekt een lijnstuk naar punt HO,VE. CN=Ø geeft voorgrondkleur, CN=1 geeft achtergrondkleur.

GOSUB 65Ø (tekst op het grafische scherm)

Print SR$ op het grafische scherm in kleur CN vanaf positie HO,VE.

GOTO 95Ø (Programma-beëindiging)

Vervangt de in BASIC gebruikelijke opdrachten END en STOP.

GOTO 1ØØØ (Programmastart)

Vervangt de in BASIC gebruikelijke opdracht RUN.

A8.2 Variabelen en BASIC-opdrachten

Namen zijn maximaal twee karakters lang, het eerste karakter is een hoofdletter (de letter O is daarvoor verboden), het tweede karakter mag een hoofdletter of een cijfer zijn. Andere toevoegingen dan $ zijn verboden.

Logische variabelen zijn 'waar' of 'onwaar'. Rekenen met de eventuele getalswaarde is niet toegestaan.

Variabelen moeten eerst een waarde krijgen en dan pas gebruikt worden.

Verboden zijn de variabelen: AS, AT, FN, GR, IF, LN, PI, ST, TI, TI$, TO.

A8.3 Toegestane BASIC-opdrachten

Uitsluitend de volgende Basic-opdrachten zijn toegestaan:

ABS	
AND	N.B.: Gebruik van haakjes is in BASICODE-3 verplicht.
ASC	N.B.: Gebruik dit met voorzichtigheid, sommige computers werken met afwijkende coderingen.
ATN	
CHR$	N.B.: Gebruik dit met voorzichtigheid, sommige computers werken met afwijkende coderingen.

COS
DATA N.B.: op een DATA regel mogen alleen in te lezen variabelen staan.
DIM N.B.: gebruik van DIM is ook verplicht bij kleine arrays
EXP
FOR ... TO ... STEP ... NEXT ...
GOSUB
GOTO
IF ... THEN N.B.: ELSE is verboden
INPUT N.B.: een promptstring is niet toegestaan
INT
LEFT$
LEN
LET
LOG N.B.: berekent de natuurlijke logaritme
MID$ N.B.: niet toegestaan is iets als A$=MID$(C$,5)
NEXT N.B.: moet gevolgd worden door een variabelenaam
NOT N.B.: haakjes verplicht
ON ... GOSUB ...
ON ... GOTO ...
OR N.B.: haakjes verplicht
PRINT
READ
REM
RESTORE N.B. RESTORE gevolgd door een regelnummer is niet toegestaan
RETURN
RIGHTS
SIN
SGN
SQR
STEP
TAB N.B.: gebruik wordt niet aanbevolen. Beter subroutine 110
TAN
THEN Zie IF ..
TO Zie FOR ..
VAL

A8.4 Regelnummering

0 - 999: Standaardroutines.
1000: Verplicht van de volgende vorm:
1ØØØ A=<waarde>:GOTO 2Ø:REM programmanaam
1010 - 19999: het hoofdprogramma
20000 - 24999: onvermijdelijke subroutines met niet-toegestane opdrachten
25000 - 29999: eventueel benodigde DATA-regels
30000 - 31999: REM-regels met een korte beschrijving
32000 - 32767: REM-regels met de naam en het adres van de auteur

Regelnummers moeten bij voorkeur oplopen in stappen van 10.

Er zijn vele merken en typen computers te koop, de een nog beter dan de ander, tenminste, als je de fabrikant moet geloven. Eén ding hebben alle computers gemeen: ze kunnen absoluut niet overweg met die leuke programma's die nu juist alleen voor dat andere merk computer te koop zijn.

BASICODE is een Esperanto voor microcomputers. Computers van verschillende merken en typen kunnen nu via BASICODE juist wel elkaars programma's inlezen en verwerken, mits men zich aan de spelregels houdt. Die spelregels, samen met een pakket hulpprogramma's, vormen de inhoud van dit BASICODE-3 pakket.

Het pakket bestaat uit een boek plus een cassette. Het boek legt uit hoe alles werkt, wat men kan en hoe dat kan, en ook wat er niet kan.

De cassette bevat 10 vertaalprogramma's, geschikt voor meer dan 30 microcomputers, en drie demonstratieprogramma's in BASICODE-3.

Op kant A staan de vertaalprogramma's voor:
- Acorn BBC Modellen B, B+64K en B+128K
- Acorn Electron
- Apple II, II+, IIe en de meeste Apple-compatibles
- Commodore 64
- Exidy Sorcerer
- Alle MSX-1 en MSX-2 computers
- Philips P2000M

Op kant B staan de vertaalprogramma's voor:
- Philips P2000T
- Sinclair Spectrum en Spectrum+
- Spectravideo SV.318 en SV.328

gevolgd door drie programma's in BASICODE-3:
- Sterrenhemel
- Muziek
- Adresbestand

gevolgd door een bestandje in BASICODE-3 met allerlei nuttige adressen.

BASICODE is een ontwikkeling van hobbyisten, voor hobbyisten, door hobbyisten. Toch is BASICODE, vooral dank zij de intense samenwerking van een grote groep mensen, geen amateuristisch geheel, maar een welhaast professioneel systeem. BASICODE-3 voegt met mogelijkheden voor geluid, grafiek en bestanden nieuwe dimensies toe aan het al vele jaren bekende concept. Uitzendingen op radio en TV, in binnen- en buitenland, maken duidelijk dat BASICODE alom een grote vlucht heeft genomen.

ISBN 90 201 1949 4

OVERZICHT BASICODE-3 SUBROUTINES

10 eerste regel van het subroutineblok
20 programmastart, systeemreset, variabelen wissen, enz. ..

N.B.: regel 10 en regel 20 kunnen niet vanuit een programma worden aangeroepen.

100 schakel om naar tekstbedrijf en wis het scherm
110 verplaats de cursor naar positie HO,VE
120 registreer de cursorpositie in HO,VE
150 print op opvallende wijze drie spaties;SR$;drie spaties
200 geef een eventueel ingedrukte toets in IN$ en IN
speciale codes: 28 = cursor links 29 = cursor rechts
30 = cursor omlaag 31 = cursor omhoog
127 = wis/delete
verder geldt altijd: 32<=IN<=95
210 wacht tot toetsindruk en geef deze in IN$ en IN
220 geef in IN de code van wat op schermpositie HO,VE te zien is
N.B.: zie de toevoeging bij subroutine 200
250 geef een piepje als attentiesein
N.B.: zie de toevoeging bij subroutine 200
260 geef een randomgetal in RV zodat Ø <= RV < 1
270 doe 'garbage collect' en geef in FR het totaal aantal vrije bytes
280 schakel de stoptoets in (FR=Ø) of uit (FR=1)
300 geef in SR$ de tekstvorm van SR, zonder spaties
310 geef in SR$ de tekstvorm van SR, geformatteerd conform CT en CN
330 verander alle kleine letters in SR$ in hoofdletters
350 stuur SR$ naar de printer
360 sluit de printerregel af met wagenterug en papieropvoer
400 maak een toon volgens SP, SD en SV
SP tooncode: 0=uiterst laag, 60='Centrale C', 127=uiterst hoog
SD toonduur in stappen van 0,1 seconde
SV volume: 0=stil, 7=normaal, 15=maximum
450 wacht maximaal SD*0,1 seconden op een toetsindruk
na afloop: toets in IN en IN$ conform de toevoeging bij subr. 200
SD verlaagd tot moment van toetsindruk of SD=Ø

500 open bestand NF$ volgens code NF
NF=even: invoer,   NF=oneven: uitvoer
NF= 0 of 1 : BASICODE-cassette
NF= 2 of 3 : eigen systeemgeheugen
NF= 4 of 5 : diskette
NF= 6 of 7 : diskette

540 lees in IN$ uit geopend bestand NF en in IN de status
IN=0: alles OK, IN=1: einde bestand, IN=-1: foutcode

560 voer SR$ uit naar het geopende bestand met code NF

580 sluit het bestand met code NF af

600 schakel om naar grafisch bedrijf en wis het scherm

620 plot het punt op positie HO,VE in kleur CN

630 trek een lijn naar punt HO,VE in kleur CN
N.B.: Ø<=HO<1 en Ø<=VE<1

650 print SR$ als tekst vanaf grafische positie HO,VE

950 beëindig het programma en schakel de machine naar normaal bedrijf

REGELNUMMERS IN BASICODE-3

De voorgeschreven indeling voor de regelnummers in BASICODE-3 is:

0 - 999: De standaardroutines (zie hoofdstuk 4).

1000: verplicht van volgende vorm:
`1ØØØ A=<stringruimte>:GOTO 2Ø:REM programmanaam`

1010 - 19999: het hoofdprogramma;

20000 - 24999: eventuele machine-afhankelijke subroutines;

25000 - 29999: DATA-regels;

30000 - 31999: REM-regels met korte programma-beschrijving en eventuele literatuurverwijzingen;

32000 - 32767: REM-regels met de naam en het adres van de maker.